

Journal of Computational Astronomy & Astronomical Computing

Revista de astronomía computacional y cálculo astronómico

Nº 3. Julio 2025

In this issue:

Editorial

- 5 **La esfera armilar** César González Crespán
- 27 **Estimación de los contactos C_1 y C_4 durante un eclipse solar parcial mediante análisis geométrico de vídeo astronómico** Javier de Elías
- 39 **Caracterización de cúmulos abiertos con Clusterix 2.0** Joaquín Álvaro
- 53 **Diseño, construcción y programación de dispositivos astronómicos: Construcción física, electrónica y firmware** Sergio Alonso y Javier Flores
- 65 **Astronomical Computing: Cálculo de efemérides** Tomás Alonso
- 79 **Software for Photometry & Astrometry: Fotometría con Tycho: fotometría de cometas Multiapertura y AfRho a partir de un Track Stack** Ramón Naves y Montse Campàs

Edited by **Federación de Asociaciones Astronómicas de España** in Madrid (c/Serrano, 117)

The Journal of Computational Astronomy & Astronomical Computing is an effort by the FAAE - Grupo de Cálculo Astronómico (GCA) to encourage the use of software tools and the development of codes and algorithms for astronomical applications within the framework of amateur astronomy, as well as to connect the amateur community with the professional astronomy community and promote cross-collaboration and ProAm projects between both groups.



Sociedad Española de
Astronomía



Federación de Asociaciones
Astronómicas de España



This work is licensed under a Creative
Commons Attribution 4.0 License



Journal of Computational Astronomy and Astronomical Computing

An online publication of Federación de Asociaciones Astronómicas de España, Grupo de Cálculo Astronómico

Editorial Board: Enrique Velasco, Sergio Díaz, Joaquín Álvaro, César González, José Luis Navarro, Enrique Sánchez, Roberto Ruiz, Ramón Naves y Montse Campàs.

Advisory Board: Luis Mederos and Matilde Fernández.

<https://federacionastronomica.es/index.php/recursos-astronomicos/journal-of-ca-ac>

EDITORIAL

En el presente número de JCAAC seguimos dando continuidad a las secciones abiertas en los números anteriores. En particular, se ofrece el segundo artículo dedicado al Observatorio Virtual, con una aplicación a la caracterización de cúmulos abiertos utilizando la herramienta Clusterix 2.0, un producto que resulta de la colaboración entre diferentes instituciones de investigación españolas. Asimismo, se presenta la segunda entrega de la sección sobre Programación de Dispositivos Astronómicos y, en la sección sobre Astronomical Computing, se trata el asunto de cómo reducir la posición de objetos astronómicos y organizar algunos cálculos. Finalmente, la sección Software for Photometry & Astrometry continúa con la aplicación del software Tycho al difícil asunto de la fotometría de cometas.

Como artículos de aplicación, contamos en este número con dos interesantes aportaciones. En la primera se presenta una esfera armilar construida en madera, y se sientan las bases del modo de funcionamiento del instrumento, con medidas reales. Además, se pone este instrumento en un contexto histórico, con interesantes explicaciones sobre la evolución del aparato y su uso en la antigüedad. En la segunda aplicación, de gran actualidad como prolegómeno de la serie de eclipses que podremos observar próximamente desde la península ibérica, se presenta un método para el cálculo de los instantes de contacto primero y cuarto en un eclipse solar parcial, con medidas reales. Los resultados son excelentes y están sujetos a mejoras, las cuales serán objeto de futuras contribuciones a JCAAC.

In the current issue of JCAAC we continue the sections introduced in previous editions. In particular, this issue features the second article devoted to the Virtual Observatory, with an application focused on the characterization of open clusters using the tool Clusterix 2.0, a product resulting from collaboration between various Spanish research institutions. In addition, we present the second installment of the section on Programming for Astronomical Devices, and in the Astronomical Computing section we address how to reduce the position of astronomical objects and perform certain calculations. Finally, the Software for Photometry & Astrometry section continues with the application of the Tycho software to the challenging task of comet photometry.

As for applied articles, this issue includes two interesting contributions. The first presents a wooden armillary sphere and lays out the basic principles of how it works, supported by real measurements. The instrument is also placed in a historical context, with insightful explanations about its evolution and use in antiquity. The second applied article is particularly timely, as a prelude to the upcoming series of eclipses visible from the Iberian Peninsula. It presents a method for calculating the first and fourth contact times in a partial solar eclipse, based on real measurements. The results are excellent and open to further improvement, which will be the subject of future contributions to JCAAC.

E. Velasco, July 2025

CONTENTS

Editorial

Applications

<i>La esfera armilar</i> , César González Crespán	5
<i>Estimación de los contactos C_1 y C_4 durante un eclipse solar parcial mediante análisis geométrico de vídeo astronómico</i> , Javier de Elías	27
Section: Herramientas del observatorio virtual , Joaquín Álvaro Contreras	
<i>Caracterización de cúmulos estelares abiertos con Clusterix 2.0</i>	39
Section: Programación de dispositivos astronómicos , Sergio Alonso y Javier Flores	
<i>Diseño, construcción y programación de dispositivos astronómicos: Construcción física, electrónica y firmware</i>	53
Section: Astronomical Computing , Tomás Alonso	
<i>Cálculo de efemérides</i>	65
Section: Software for Photometry & Astrometry , Ramón Naves y Montse Campàs,	
<i>Fotometría con Tycho: fotometría de cometas multiapertura y AfRho a partir de un Track Stack</i>	79

All contents of this publication are licensed under a Creative Commons Attribution 4.0 License
Published codes can be used and distributed under a Gnu gpl license



With the sponsorship of Sociedad Española de Astronomía (SEA), Grupo ProAm

APPLICATIONS

La esfera armilar

César González Crespán¹¹Asociación Astronómica de Vigo, Spain. E-mail: cgcrespan@gmail.com.**Keywords:** esfera armilar, astrolabio, meteoroscopio, zodíaco, eclíptica© Este artículo está protegido bajo una licencia [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/)Este artículo adjunta un *software* accesible en <https://github.com/JCAAC-FAAE>

Resumen

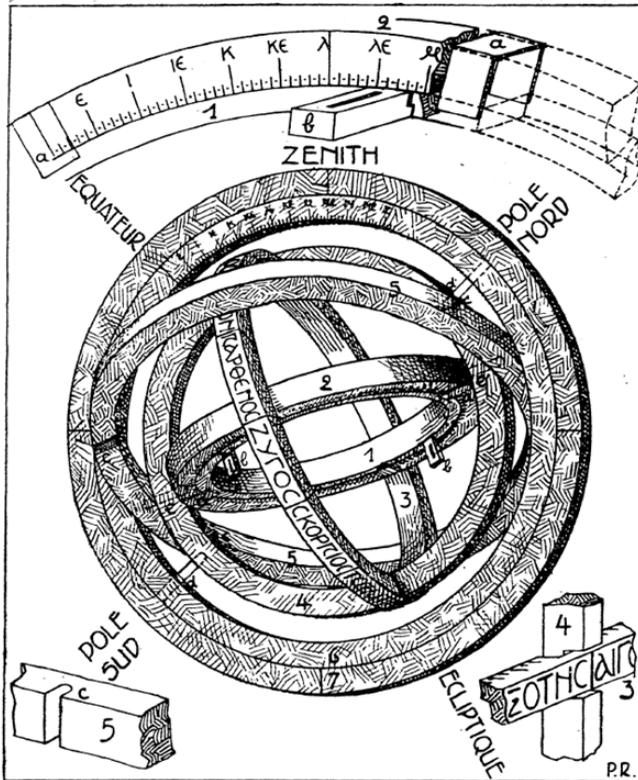
Con una réplica de la esfera armilar eclíptica, construida en madera, se realiza una descripción práctica del instrumento, como debe prepararse, y orientarse, para realizar la observación del Sol, la Luna y otros astros. Se utiliza para obtener coordenadas celestes, longitudes y latitudes eclípticas. Previamente se compara con la esfera armilar ecuatorial, cuyo fin es obtener ascensiones rectas y declinaciones de los astros, y con otro instrumento, llamado meteoroscopio. Se analizan varios estudios realizados sobre estos instrumentos, determinando que su nombre es astrolabio (buscador de estrellas) esférico. Se explica la evolución de estos instrumentos astronómicos a lo largo de los tiempos, y sus aplicaciones, así como su relación con el Astrolabio plano, y el telescopio.

Abstract

Using a wooden replica of the ecliptic armillary sphere, a practical description of the instrument is provided, including how it should be prepared and oriented for observing the Sun, Moon, and other celestial bodies. It is used to obtain celestial coordinates, ecliptic longitudes and latitudes. It is compared, previously, with the equatorial armillary sphere, whose purpose is to obtain right ascensions and declinations of celestial bodies, and with another instrument called meteoroscope. Several studies on these instruments are analyzed, determining their name as a spherical astrolabe (star finder). The evolution of these astronomical instruments over time and their applications are explained, as well as its relationship with the flat astrolabe and the telescope.

1. Introducción

Ya en el siglo IV a.C. era conocida la Esfera armilar, como instrumento astronómico para la observación de los astros, por el griego Eudoxo, aunque tal vez fuera conocida en tiempos anteriores, en Babilonia. De esta manera era posible determinar las coordenadas celestes, ascensiones rectas y declinaciones, que estaban referidas al plano ecuatorial. También era conocido el sistema de referencia fundamentado en el horizonte, estableciendo los puntos cardinales, así como las alturas y los azimutes. Se produjo un avance con la invención de la esfera armilar eclíptica, que se fundamentaba en un sistema de referencia asociado al plano eclíptico, que explicaba de una forma uniforme el movimiento de los astros. Hiparco tuvo que ver en la invención de este aparato [1], también en la creación del astrolabio plano, con la proyección estereográfica. Claudio Ptolomeo dejó escrito, en el *Almagesto* un catálogo de estrellas calculado por medio de la observación, con un instrumento armilar, que llama astrolabio, dando coordenadas eclípticas. Fue utilizado después, por astrónomos griegos, árabes y europeos, hasta llegar a Copérnico, Brahe y Kepler.



Reconstitution schématique de l'astrolabe, par M. P. Rome, ingénieur-architecte.

Figura 1. Esfera armilar eclíptica - Astrolabio esférico. Reconstrucción de Rome (1927).

2. Reconstrucción del astrolabio esférico por Rome (1927)

Rome realiza [2], en 1927, una reconstrucción del astrolabio esférico, a partir de los textos de Pappus de Alejandría (s. IV d.C.): Anillo 1 que lleva las miras bb. Anillo 2 perpendicular a la eclíptica (graduado; girando sobre los ejes ee; llevando el anillo 1 retenido por bridas). Anillo 3, zodíaco graduado (soldado a 4). Anillo 4, coluro solsticial (soldado a 3; atravesado por los ejes ee que llevan los anillos 2 y 5; sujetado en los ejes dd que le ligan al anillo 6). Anillo 5, perpendicular a la eclíptica (girando sobre los ejes ee; no graduado; para poder ir hasta el plano meridiano, debe tener cuatro ranuras cccc en las cuales vienen entonces a insertarse los ejes dd). Anillo 6, meridiano de reglaje (graduado de 0° a 63°; atravesado por los ejes dd; retenido al siguiente). Anillo 7, meridiano fijo (llevando cuatro trazos que marcan la vertical y la horizontal).

Tiene un anillo, nº 5, longitud eclíptica exterior, girando alrededor de los polos eclípticos, de movilidad limitada hasta los pernos que sujetan el eje de rotación celeste, polos norte - sur. La configuración del instrumento se complica un poco, pero este anillo tiene utilidad, ya que así se dispone de dos anillos para marcar las longitudes eclípticas, uno tangente al astro de referencia, permaneciendo intocable en posición de equilibrio, otro que se sitúa tangente al astro cuya longitud eclíptica se desea medir. El aro 5 parece posible situarlo por el interior del aro 4, en vez de, por el exterior, de esta forma, no chocaría con los pernos del eje celeste, y tendría mayor amplitud de movimiento, sin limitaciones.

Jarosław Włodarczyk (1985) realizó [3] mediciones con la Esfera Armilar Eclíptica del Instituto de Astronomía Geodésica de la Universidad Técnica de Varsovia, construida en la década de 1950, a partir de este modelo.

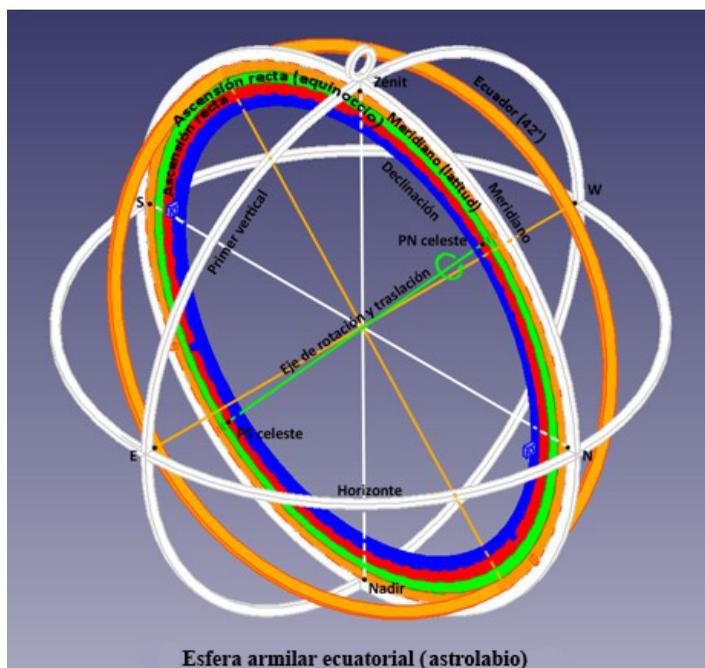


Figura 2. Esquema de la esfera armilar ecuatorial – astrolabio esférico.

3. Dos tipos de esferas armilares

3.1. La esfera armilar ecuatorial - astrolabio esférico

Es un instrumento con un sistema de referencia fundamentado en el horizonte, y otro sistema de referencia fundamentado en el plano ecuatorial, que permite determinar coordenadas celestes ecuatoriales, calculadas desde ese plano, ascensión recta y declinación. Es un astrolabio (buscador de estrellas) esférico.

La esfera armilar ecuatorial era conocida desde el siglo IV a.C., probablemente antes, en Babilonia. Los aros de color blanco representan el horizonte, el plano meridiano, perpendicular al anterior, al que corta, determinando los puntos cardinales norte y sur, y el vertical primario, plano perpendicular a los dos anteriores, determinando, en su intersección con el horizonte, los puntos cardinales este y oeste. También corta al meridiano en los puntos zénit y nadir. Es un sistema de referencia basado en el horizonte, que constituye un triedro estático, en el centro del cual está la Tierra. Un aro interior al meridiano, que es movable circularmente a lo largo de este, sujeto por bridas, contiene al eje de rotación terrestre. Se coloca de forma que el eje de rotación forme un ángulo con el plano horizontal igual a la latitud del lugar, altura del polo, y queda también estático, de forma que un aro perpendicular, solidario al mismo, queda paralelo al ecuador terrestre, y estático también. El eje de rotación terrestre se marca en el meridiano según la latitud del lugar de observación. Otro aro interior, que gira alrededor del eje de rotación, determinado por dos cilindros que encajan en dos huecos del aro anterior, determina el origen de las ascensiones rectas, apuntándose al punto equinoccial, si se sabe encontrarlo, al sol o la luna, para tomar en estos casos, la diferencia de ascensiones rectas. Otro aro interior, que gira también alrededor del eje de rotación, se apunta al astro del cual se quiere determinar la ascensión recta, y otro aro interior, sujeto por bridas, que se desliza circularmente por la parte interior, determina la declinación, ángulo por encima o debajo del ecuador, por medio de unas miras que se dirigen al astro a medir.

El problema que presenta este instrumento es que confunde los ejes de rotación y traslación en un

mismo eje, con lo cual las ascensiones rectas, que determina, no tienen una variabilidad uniforme con el tiempo, que se pueda calcular fácilmente. Los “Comentarios al Aratus latinus” son el único texto que se conserva de Hiparco [4]. Arato era un poeta que escribió “Los Fenómenos”, un poema sobre el catálogo de estrellas de Eudoxo, astrónomo que vivió en el siglo IV a.C. Presenta los límites de algunas constelaciones, por lo que sólo da la ascensión recta o la declinación de algunas estrellas, pero no las dos coordenadas juntas. Estas coordenadas fueron obtenidas con una esfera armilar ecuatorial.

3.2. La esfera armilar eclíptica - astrolabio esférico

A alguien, en el siglo II a.C., probablemente a Hiparco, se le ocurrió sustituir el aro verde de la figura anterior, por un aro llamado revólver, que representa al plano llamado coluro solsticial, que contiene los dos solsticios, los dos polos celestes, y los dos polos eclípticos, que gira igualmente alrededor del eje de rotación, pero añadiendo un nuevo eje, de traslación, separado del anterior por $23,5^\circ$, la oblicuidad, que pasa por los polos eclípticos, alrededor del cual gira un aro interior, en rojo, llamado astrolabio, que determina las longitudes eclípticas, a medir desde el punto al cual se apunta, que puede ser el punto equinoccial, el sol, o la luna, determinadas en estos dos últimos casos por diferencia. Otro aro interior, que se desplaza por dentro, sujeto por bridas, determina la latitud eclíptica, entre 0° y 90° , boreal, o entre 0° y -90° , austral. Perpendicular al aro que representa al coluro solsticial, y solidario a él, hay otro aro, que representa al plano eclíptico, o zodíaco, con las doce dodecatemorias, o signos zodiacales, cada una de 30° , hasta completar los 360° de la circunferencia. El sol, dinámico, se mueve a lo largo del zodíaco, en dirección de las longitudes eclípticas crecientes. La línea de los equinoccios retrocede en dirección de las longitudes eclípticas decrecientes, con el consiguiente aumento de las longitudes eclípticas de los astros. Los aros del sistema de referencia terrestre son iguales a la esfera anterior, permaneciendo estáticos, con la tierra en el centro, también estática. El anillo superior, llamado “gancho”, permite suponer que, para ciertos propósitos, el aparato se utilizaba colgado, alcanzada la horizontalidad por la propia inercia de su peso, aunque lo normal era que se apoyara sobre un pedestal.

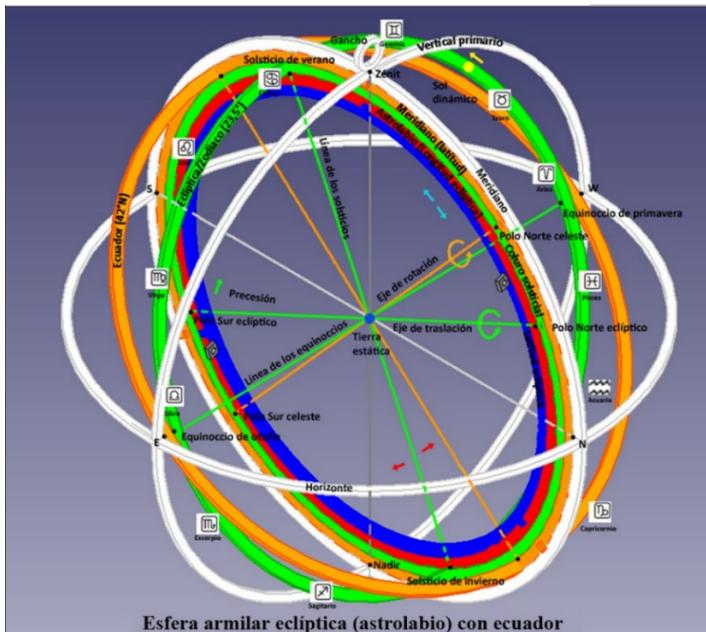


Figura 3. Esquema de la esfera armilar eclíptica – astrolabio esférico, con ecuador.

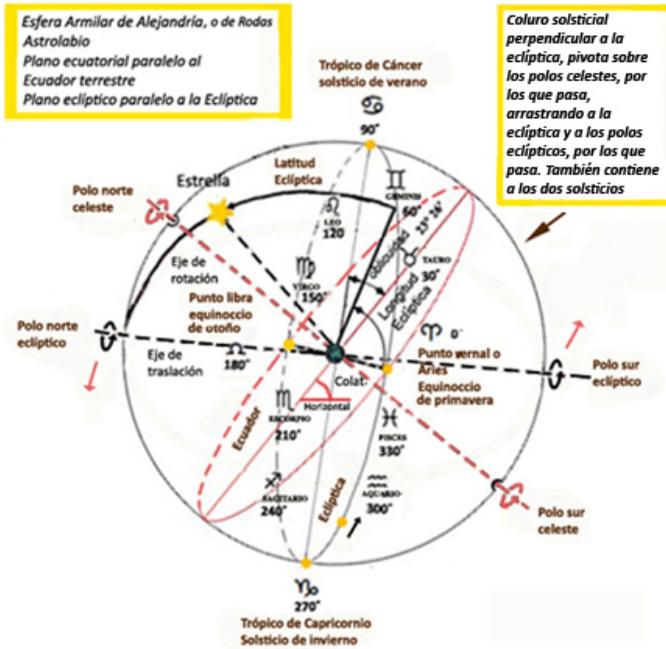


Figura 4. Mecánica de la Esfera armilar eclíptica (Astrolabio esférico).

4. El meteoroscopio

4.1. Sobre el meteoroscopio

De acuerdo con la reconstrucción del meteoroscopio, a partir del texto encontrado en un palimpsesto, “Recuperado el tratado de Ptolomeo sobre el meteoroscopio” [5], de Victor Gysembergh, Alexander Jones, Emanuel Zingg, Pascal Cotte, Salvatore Apicella, [Archive for History of Exact Sciences (2023)], el instrumento descrito presenta características híbridas entre un instrumento de observación celeste ternario, fundamentado en el plano eclíptico, anillos 5, 6 y 7, y otro instrumento que no es de observación, sino de representación terrestre, fundamentado en el plano ecuatorial, anillos 8 y 9. Los anillos exteriores, 1, 2, 3, 4 y Z, constituyen un sistema de referencia fundamentado en el horizonte, y son comunes, y necesarios, para las dos funcionalidades. Para los antiguos astrónomos, la Tierra permanecía estática en el centro del sistema de referencia horizontal, y el Sol era dinámico, girando alrededor de la Tierra, en el centro de dicho sistema. La configuración, en esta figura, de las dos primeras funcionalidades tiene sus limitaciones, ya que los movimientos de un sistema y otro llegan a entrecruzarse, y sus ejes de giro interfieren entre sí. El anillo 7, astrolabio, recorre la eclíptica, girando en torno a los polos eclípticos Nz-Sz, hasta los pernos que caracterizan al eje de rotación terrestre Ne-Se, y no puede seguir más allá su movimiento, sólo retroceder. El anillo 8, “vertical”, girando sobre los puntos del revolver que representan los polos celestes, puede representar un meridiano arbitrario de cualquier localidad, es perpendicular al ecuador. El meridiano local está representado en el sistema de referencia fundamentado en el horizonte, por el anillo 4, luego podría utilizarse para marcar una longitud terrestre arbitraria. El anillo 9, “totalmente basculante”, girando sobre los puntos del anillo vertical que representan las intersecciones del meridiano arbitrario con el ecuador, su nombre refleja el hecho de que se puede colocar en cualquier plano, este anillo puede representar **un plano perpendicular** al horizonte (el **vertical primario**) para cualquier localidad.

Con este instrumento **no se puede tomar la altura del polo boreal** (que es igual a la latitud) **durante la noche o el día**, cuando no se ve el polo, por la brillante luz del Sol, ya que, para su reglaje, es necesario conocerla. De la misma manera que la esfera armilar eclíptica, con la que comparte algunas características, este instrumento permite representar la posición del meridiano, aros 1 y 4, en relación a la eclíptica, aro 6, determinada por las posiciones del Sol, que sigue la eclíptica, y de la Tierra. Para ello, el aro eclíptico del instrumento se debe poner paralelo a la eclíptica, apuntándolo al Sol, de manera que la parte anterior, haga sombra sobre la parte interior cóncava. También dice, en la Geografía, “sirve para. . . , así como el arco del ecuador, comprendido entre dos meridianos, y de la misma manera los arcos de los paralelos al ecuador”.

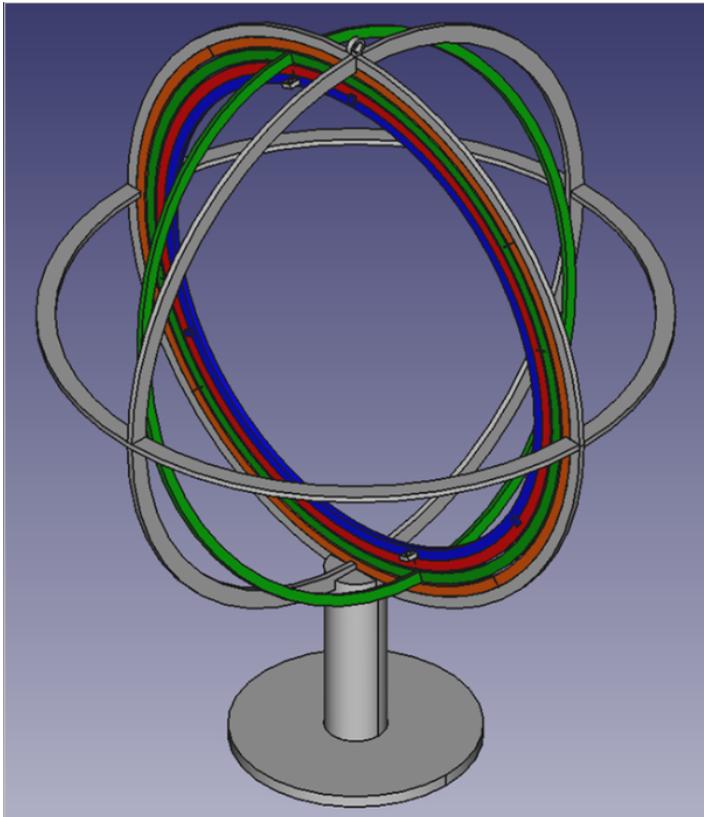


Figura 5. *Diseño, con Freecad, de una esfera armilar eclíptica (Astrolabio esférico), de ocho anillos, con gancho y pedestal.*

Al comienzo del capítulo III, Libro 1º, de la “Geografía” (ver más adelante el capítulo completo), donde describe el meteoroscopio, que, como hemos visto, tiene elementos que son comunes a la Esfera Armilar Eclíptica (Astrolabio esférico), Claudio Ptolomeo dice “AQUELLOS que nos precedieron. . .”. Es decir, fueron sus antepasados, quienes inventaron, tanto el Meteoroscopio, como la Esfera Armilar Eclíptica. Luego ninguno de estos dos aparatos puede atribuirse a Claudio Ptolomeo, sino a algún astrónomo que le precedió.

El ángulo Katabatico Descendente. K1 Podría ser la colatitud ($90 - \text{latitud}$) de un lugar arbitrario.

El ángulo Antiscio Que por estar sobre el mismo meridiano y en el hemisferio opuesto, proyecta al

mediodía la sombra en dirección contraria. Que distan igualmente de los trópicos. a podría ser la diferencia de longitud entre el meridiano local y un lugar arbitrario.

4.2. Sustitución de los anillos nºs 8 y 9 de “Recuperado el tratado de Ptolomeo sobre el meteoroscopio”

El anillo nº 8, de sustitución, para reemplazar a los anillos números 8 y 9 anteriores, sería de la siguiente forma:

8. Las latitudes eclípticas se calculan desde el plano eclíptico, perpendicularmente al mismo, siguiendo el círculo de la anterior armilla de longitud, hacia el polo norte, o sur, eclípticos, con unas miras diametralmente opuestas, situadas en un aro que se desliza por el interior del astrolabio, y se mantiene en el plano del astrolabio sujeto por bridas, de la misma manera que el cuarto anillo, hasta apuntar a la estrella a medir. El instrumento resultante sería de las mismas características que el de Rome, aunque sin el anillo longitud eclíptica exterior. Tendríamos una esfera armilar eclíptica, astrolabio esférico. Por tanto, la esfera armilar eclíptica, tiene un parecido muy grande, compartiendo la mayor parte de los anillos, al meteoroscopio.

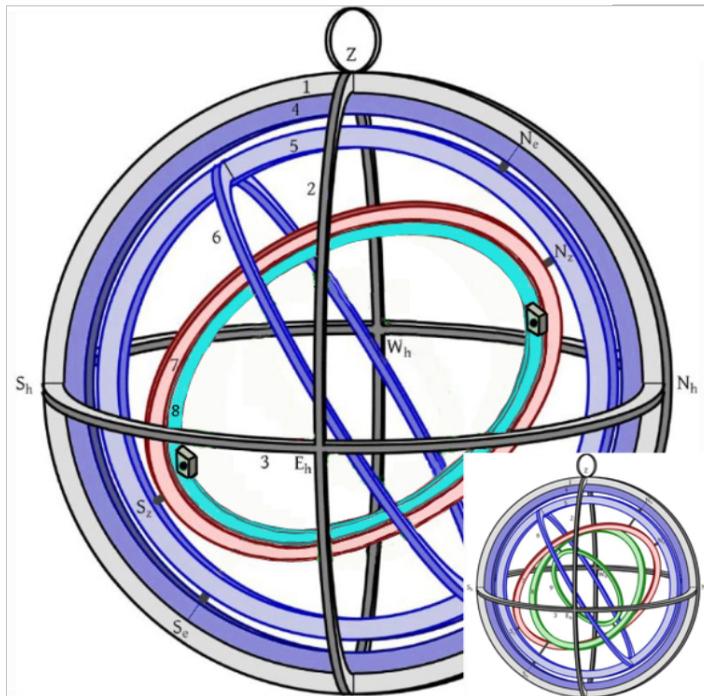


Figura 6. Esfera armilar eclíptica – Astrolabio esférico. Modificado de Victor Gysembergh, Alexander Jones, Emanuel Zingg, Pascal Cotte, Salvatore Apicella, “Recuperado el tratado de Ptolomeo sobre el meteoroscopio”. Archive for History of Exact Sciences (2023). La figura principal de esta figura es un astrolabio esférico, no un meteoroscopio, que es el que está en la imagen a escala reducida, abajo a la derecha.

5. Construcción de una esfera armilar eclíptica

Se ha construido una réplica del instrumento esfera armilar eclíptica – astrolabio esférico, en madera, de 8 mm de grosor, cortada con láser, cuyos anillos exteriores tienen 50 cm de diámetro mayor. Se toma una oblicuidad de $23,5^\circ$, para situar los ejes celestes de rotación ecuatorial, y traslación eclíptica, separados esa distancia angular en el anillo revólver, o coluro solsticial. Para su diseño se han tomado como base los estudios que se comentan, en particular el de Rome y el de Gysembergh & al., sobre el meteoroscopio, por medio de un programa de diseño asistido por ordenador, Freecad. Los mayores problemas que se encontraron han sido el ensamblaje de los aros, de adentro hacia afuera, para lo cual hubo que cortar los exteriores, con uniones provisionales, en cola de milano. Otro problema encontrado ha sido la alineación de los taladros de los cantos, por carecer de una herramienta adecuada, para alinear de forma perfecta los mismos. Los ejes de giro se han resuelto con cojinetes constituidos por remaches de aluminio de 4 mm, con el agujero algo agrandado para que puedan girar sin problema.



Figura 7. Esfera armilar eclíptica – Astrolabio esférico. Construida en madera.

6. Instalación y orientación de los instrumentos esferas armilares

Para colocar la esfera armilar en posición correcta para la observación:

1. Alinear el instrumento al plano del meridiano local
2. Ajustar el cenit del instrumento al cenit verdadero
3. Ajustar el instrumento a la latitud geográfica del lugar

El instrumento se coloca de manera que el anillo meridiano quede paralelo al plano meridiano local. Podemos utilizar un compás magnético, corrigiendo la declinación. Sirve la del teléfono móvil, eligiendo la opción Norte verdadero. En tiempos antiguos se trazaba una línea en el suelo del observatorio que marcaba el meridiano, que se utilizaba para localizar el mismo. Para obtener la dirección de esta línea se ponía un gnomon vertical en el suelo, y se iba marcando el punto de la sombra más corta del día, que correspondía al momento en que el sol cruzaba el meridiano local. Uniendo la base del gnomon con el punto que marcaba la sombra más corta, se obtenía la línea meridiana. Se podía repetir esta operación todos los días de un año, de forma que las sombras eran más largas hacia el solsticio de invierno, y más cortas hacia el solsticio de verano. Las sombras eran de igual longitud, hacia la mitad, en los dos equinoccios.



Figura 8. Correctamente nivelado, orientado y apuntado el instrumento al Sol, en 352° , piscis, 11 de marzo de 2025, 12:19 p.m.CET. La parte delantera del anillo eclíptico hace sombra sobre la parte interior cóncava de mismo anillo eclíptico. El anillo eclíptico queda paralelo al plano eclíptico.

El cénit verdadero debe quedar apuntado al cénit real. Para ello, poner el instrumento perpendicular al plano del horizonte. Se puede ajustar el instrumento utilizando una plomada que verifique el eje vertical del instrumento, lanzada desde el punto cenital del anillo meridiano hasta el punto del nadir. Si el instrumento está bien construido, se puede poner sobre una mesa alta (de aprox. un metro de altura), columna, o trípode, y utilizar un nivel de burbuja para alcanzar la horizontal, con calzos. De esta manera, el instrumento queda situado a la altura de la cara del observador, para poder realizar los reglajes del aparato, y observar a través de las miras de una forma cómoda, agachándose ligeramente.

Se coloca, entonces, el anillo meridiano/latitud, por dentro del anillo meridiano, de manera que el punto latitud coincida con la marca de la escala correspondiente a la latitud geográfica del lugar de observación, en este caso la de La Ramallosa ($\varphi = 42.12^\circ$).

Las mediciones con el instrumento han de realizarse rápido, ya que el anillo eclíptico se desajusta pronto con respecto al plano eclíptico. En 4 minutos se desajusta un grado ($360^\circ/24 \text{ h.}$).

7. Apuntamiento y medición con la esfera armilar eclíptica

7.1. Apuntamiento del instrumento esfera armilar eclíptica - astrolabio esférico y medición

Para apuntar el instrumento se busca la posición del anillo eclíptico para la que la parte delantera, orientada hacia el sol, haga sombra sobre la parte interior cóncava del mismo anillo. Esta posición es relativamente fácil de obtener, se ve claramente, y es única. En ese momento, el Sol está en la longitud eclíptica que le corresponde en la escala de longitudes eclípticas del anillo eclíptico, y se puede leer desde el lado contrario, estando en ese momento el anillo eclíptico paralelo al plano eclíptico. El anillo longitud eclíptica (interior), en ese momento, y si se apunta hacia el sol, también debe proyectar sombra sobre sí mismo, por la parte interior cóncava. Por la configuración de los anillos, y las restricciones que tienen sus movimientos sujetos por los ejes de giro, no es posible que el anillo coluro solsticial proyecte sombra sobre el anillo eclíptico, en ninguna posición.

Una vez, con el anillo eclíptico paralelo al plano eclíptico, se está en condiciones de medir, con el anillo de longitud eclíptica, que gira alrededor del eje eclíptico, la longitud eclíptica de la Luna. Corriendo posteriormente las miras, hacia el norte eclíptico, si la Luna tiene latitud boreal, o hacia el sur eclíptico, si la Luna tiene latitud austral, podemos obtener la latitud eclíptica de la Luna, cuando las miras estenopeicas, por el lado izquierdo del anillo, están alineadas tangencialmente hacia el limbo oeste del disco lunar, que usaremos como referencia. La operación anterior debe realizarse durante el día, cuando

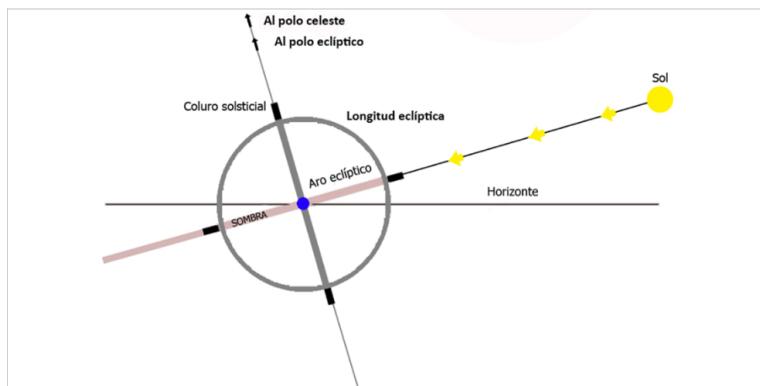


Figura 9. Apuntamiento del astrolabio (Claudio Ptolomeo, *Almagesto*): Aro Eclíptico hace sombra sobre la parte interior cóncava de sí mismo.

se ve el Sol, pero también debe verse la Luna, que se utilizará como referencia después, con la corrección adecuada, para poner el anillo eclíptico paralelo al plano eclíptico, una vez que el Sol se ha puesto, y está bajo el horizonte, con lo cual se pierde la posibilidad de utilizarlo directamente como referencia. Para observar el Sol y la Luna, juntos, tantas veces como sea posible, la Luna puede estar situada al este del Sol, es decir, sale después, o está en fase creciente, para ocultarse bajo el horizonte posteriormente al Sol. En este caso el cielo se oscurece al ponerse el Sol, y poco tiempo después se pueden observar otros astros, tomando como referencia la Luna. Si la Luna se pone antes, es decir, está al oeste del Sol, en fase menguante, podríamos utilizarla también como referencia, pero tenemos que esperar más horas, hasta que se eleve sobre el horizonte, por el este, antes que salga el Sol, antes del amanecer.

Si la Luna está cerca del Sol, es de poca utilidad realizar la observación, puesto que se va a mantener por encima del horizonte, poco tiempo después que el Sol, en el crepúsculo, cuando se ven pocas estrellas, debido a la luz del Sol. Es conveniente que hayan pasado dos o tres días desde la Luna nueva, o que falten varios días para la misma.

De acuerdo con Ptolomeo, repetimos, durante el día, que se puedan ver los dos astros simultáneamente, la medición de la longitud eclíptica de la Luna, con respecto al Sol, tantas veces como sea posible, por ejemplo, cada media hora, cada hora o cada dos horas. La razón es que hay que determinar la tendencia de la Luna en longitud eclíptica a lo largo del día de medición. Tengamos en cuenta que la Luna se mueve rápidamente en velocidad angular, de modo que recorre 360° de circunferencia en unos veintinueve días y medio, lo que significa que en un día se desplaza algo más de doce grados. Por tanto, una vez que se haya ocultado el Sol, y ya no pueda ser usado como referencia, por no ser visible, tenemos que prever, por medio del cálculo, la longitud eclíptica teórica de la Luna, correspondiente a la hora que se realiza la medición. La corrección sólo aplica a la longitud eclíptica, puesto que la otra componente del movimiento lunar, la latitud eclíptica, está predeterminada por la anterior, y no se usa como corrección.

Durante la noche también se puede usar como referencia una estrella de longitud eclíptica conocida, ajustando los aros de longitud eclíptica y zodiaco, o los planetas Venus, Marte, Júpiter o Saturno, calculando previamente su longitud eclíptica con respecto al sol, por lo tanto, deben ser visibles durante el día, antes de que se ponga el Sol, es decir, debe estar el astro situado al oriente del Sol. El instrumento se desalinea rápidamente, un grado en cuatro minutos, por lo tanto, se debe realizar la medición rápidamente. En caso contrario, hay que volver a alinearlo. El Sol recorre los 360° de la circunferencia en un año, es decir, tarda $365\frac{1}{4}$ días en recorrer la eclíptica, por lo que varía su longitud eclíptica en un valor ligeramente inferior a $1^\circ/\text{día}$. La precisión del instrumento es, según Włodarczyk, de $1/10^\circ$. Las estrellas situadas en latitudes eclípticas próximas al plano eclíptico son más difíciles de medir, hay una zona muerta por el grosor del aro eclíptico, que impide verlas con precisión, y apuntar las miras.

El instrumento es muy sensible al viento. Si está bien cortado, y construido, con unos sencillos cojinetes metálicos de aluminio de refuerzo de los ejes de giro, se equilibra solo. El mayor problema son los taladros de los cantos, que deben estar bien alineados, y centrados. En caso contrario los aros no se tienen estables.

7.2. Apuntamiento de la esfera armilar eclíptica - astrolabio esférico tras la puesta de Sol

El astrolabio planisférico sirve como reloj, ya que da las horas equinocciales (24 horas), a partir de la longitud eclíptica del sol en el día de medición, que está en la parte trasera, en la relación de escalas $360^\circ/365\frac{1}{4}$ días, marcándola con la regla en el círculo eclíptico oblicuo de la parte delantera, y la altura del mismo, durante el día, para obtener horas del día, o la de una estrella, durante la noche, para obtener horas de la noche, que figure en la araña del astrolabio planisférico, y que sea visible, y se pueda medir su altura con la alidada del instrumento, llevada su posición, por la parte delantera, al almicantrat, o círculo de alturas iguales correspondiente.

Se coloca el lado izquierdo del anillo de longitud eclíptica (interior), astrolabio esférico, en la longitud eclíptica calculada para la progresión de la Luna (u otro astro), a la hora de medición, en el anillo eclíptico. Se mueven de forma solidaria los anillos de longitud eclíptica y eclíptica (zodiaco), conservando su colocación relativa calculada anteriormente, hasta que se sitúa la parte izquierda del anillo eclíptico tangente al limbo oeste de la Luna (u otro astro). En ese instante el anillo eclíptico se sitúa paralelo al plano eclíptico real, y el Sol (no visible), a la longitud eclíptica que le corresponde en el anillo eclíptico, característico de su fondo estelar. De esta manera nos encontramos en la misma situación que durante el día producía la sombra en el interior del anillo eclíptico. La latitud eclíptica no es necesario calcularla, puesto que está predeterminada (sólo hay un grado de libertad). No es necesario realizar un ajuste de la longitud eclíptica por el semidiámetro lunar, en la dirección de la componente longitud eclíptica, ya que vamos a considerar siempre, como referencia el limbo oeste lunar, en vez del centro lunar. Si utilizáramos este, si que habría que sumar el semidiámetro lunar. El valor aproximado del semidiámetro lunar (semejante al solar), es 15', aunque varía según las fechas, por el acercamiento o alejamiento del astro, debido a que la órbita es elíptica. Estaríamos ligeramente por encima del límite de precisión del instrumento. En el caso del Sol, que brilla mucho, no hay que hacer ningún ajuste por semidiámetro, ya que la correcta alineación por la sombra sobre la parte interior cóncava del aro eclíptico, garantiza que su centro está en el punto de longitud eclíptica que le corresponde, y este se toma como referencia.

Una vez realizado el procedimiento anterior, sin tocar el anillo eclíptico, para que se mantenga en la posición correcta, paralelo a la eclíptica, se mueve el anillo de longitud eclíptica hasta el astro cuya longitud eclíptica se desea calcular, se toma el dato longitud eclíptica. Posteriormente se desplaza por su interior, el anillo de latitud eclíptica, hasta ver el astro a medir a través de las dos miras estenopeicas. Se toma el dato de latitud eclíptica. En esta situación vemos que puede ser conveniente tener un segundo anillo longitud eclíptica (exterior), como en la reconstrucción de Rome, a partir de los Cuadernos de Pappus, ya que lo utilizaríamos para determinar la longitud eclíptica lunar, y moveríamos el anillo longitud eclíptica interior hacia el astro a medir para determinar longitud y latitud eclípticas, o, al contrario, intercambiando los aros.

7.3. Resumen del funcionamiento de la Esfera Armilar Eclíptica

Durante el día, se apunta el aro eclíptico del instrumento hacia el Sol, hasta que la parte anterior del mismo haga sombra sobre la parte cóncava interior. En ese momento el Sol está en la longitud eclíptica que le corresponde y marcada en la escala del aro. Con la Luna visible, al oriente del Sol, se coloca la parte izquierda del aro astrolabio tangente al limbo oeste lunar. Se toma la longitud eclíptica de la Luna, y se repite la operación varias veces, mientras sean visibles simultáneamente la Luna y el Sol. Se calcula así la tendencia en longitud eclíptica de la Luna. Tras la puesta de Sol, por medio de la posición de la Luna corregida a través del cálculo por el tiempo transcurrido, marcándola en el aro zodiaco, y colocando el aro astrolabio en esta marca, tangente al limbo oeste lunar, queda orientado el instrumento hacia el Sol, que está oculto. De esta forma, por la posición de la Luna corregida, determinamos la posición del Sol, y el aro eclíptico, vuelve a quedar paralelo al plano eclíptico real. Manteniendo estático el aro eclíptico, y moviendo el aro astrolabio hacia cualquier otro astro, se determina su longitud eclíptica, que está indicada en la escala del aro eclíptico. Para determinar la latitud eclíptica del astro, se hace deslizar el aro interior, hasta verlo a través de las miras.

De igual manera, cualquier otro astro, si es conocida su longitud eclíptica, sirve de referencia, poniendo tangente dicho astro al aro astrolabio en la marca de su longitud eclíptica en el zodiaco, para después mover el citado aro astrolabio, hasta el astro cuya longitud eclíptica se va a determinar, poniéndolo tangente, y leyendo la longitud eclíptica que marca su intersección con el aro eclíptico.

7.4. Mediciones de prueba con la esfera armilar eclíptica

Se han realizado varias mediciones de las longitudes y latitudes eclípticas del Sol, la Luna, y de varias estrellas, con el instrumento, con el fin de comprobar su correcto funcionamiento, sin pretender afinar demasiado, ya que son conocidas sus limitaciones por las carencias de la construcción, bien cortados los aros, por láser, pero con los taladros algo descentrados.

Ubicación		Longitud geográfica		Latitud geográfica		Elevación		
		8 ° 48 ' 23,61 "		42 ° 8 ' 9,5 "		10 m		
		8,81 " W		42,14 " N				
Oblicuidad		23 ° 26 ' 18,7 "		23,44 "				
Día	Hora	Longitud eclíptica			Latitud eclíptica			
			Semidiámetro (aparente)					
				Medición		Medición		
Sol	11/3/2025 La Ramallosa	12 h 19 m 0 s	351 ° 7 ' 59,5 "	351,13 "	352,0 "	- 0 ° 0 ' 6,5 "	- 0,00 "	? "
Luna	7/5/2025 Vigo	18 h 0 m 0 s	173 ° 19 ' 39,2 "	173,08 "	?	+ 0 ° 3 ' 23,1 "	+ 0,06 "	? "
Luna	7/5/2025 Vigo	23 h 0 m 0 s	175 ° 14 ' 54,1 "	175,00 "	Tangente	- 0 ° 24 ' 8,2 "	- 0,40 "	? "
Capella	7/5/2025 Vigo	23 h 0 m 0 s	82 ° 12 ' 26,6 "	82,21 "	Tangente	+ 22 ° 51 ' 56,9 "	+ 22,87 "	? "
Júpiter	7/5/2025 Vigo	23 h 0 m 0 s	82 ° 46 ' 20,2 "	82,77 "	?	- 0 ° 13 ' 26,4 "	- 0,22 "	? "
Proción	7/5/2025 Vigo	23 h 0 m 0 s	116 ° 7 ' 59,9 "	116,13 "	Tangente	- 16 ° 1 ' 34,6 "	- 16,03 "	? "
Marte	7/5/2025 Vigo	23 h 0 m 0 s	128 ° 49 ' 0,8 "	128,82 "	Tangente	+ 1 ° 51 ' 48,6 "	+ 1,86 "	? "
Regulus	7/5/2025 Vigo	23 h 0 m 0 s	150 ° 10 ' 58,9 "	150,18 "	151,0 "	+ 0 ° 27 ' 56,6 "	+ 0,47 "	? "
Spica	7/5/2025 Vigo	23 h 0 m 0 s	204 ° 12 ' 1,8 "	204,20 "	204,0 "	- 2 ° 3 ' 23,3 "	+ 2,06 "	? "
Arturo	7/5/2025 Vigo	23 h 0 m 0 s	204 ° 35 ' 37,2 "	204,59 "	Tangente	+ 30 ° 43 ' 11,1 "	+ 30,72 "	? "

Figura 10. Mediciones de longitudes y latitudes eclípticas de varios astros con la esfera armilar eclíptica. Estimaciones realizadas con Stellarium. Es posible que se añadan más mediciones reales. Ver Ref. [6].

El día once de mayo, en la Ramallosa, a las doce horas diecinueve minutos, se realizó la medición del Sol, obteniéndose el interior del aro eclíptico en sombra, en longitud eclíptica 352°, siendo la esperada, de 351,13°. El día cinco de mayo, en Vigo, se intentó realizar una medición simultánea del Sol y la Luna, en que la Luna estaba en fase creciente, octavo día. Había rachas de viento de 14 nudos. A las siete de la tarde se instaló el instrumento, pero al intentar medir la Luna, con el aro eclíptico en sombra por el interior, es decir, alineado con el plano eclíptico, una racha fuerte de viento, hizo que se perdiera la estabilidad, arrastrando al instrumento, que cayó al suelo, rompiéndose varios aros. Hubo que reconstruir el instrumento con cola de madera. Al día siguiente, seis, se volvió a intentar la medición, a las seis de la tarde, con algo menos de viento, y el instrumento, casi reparado (después comprobé que no lo estaba del todo), inicié la medición de la Luna, pero viendo que había viento, y que era fácil que entrara alguna racha como la del día anterior, suspendí la medición. El día siete, miércoles, por la tarde, a las seis, hizo un viento parecido al anterior, previsión de rachas de unos catorce nudos, luego no intenté hacer nada. Ese mismo día, con un viento máximo previsto de diez nudos, por lo tanto, suave, a las once de la noche, se intentó la medición, en una situación de calma, y sin rachas de viento. Se encontró el limbo occidental lunar, tangente al lado izquierdo del aro astrolabio, en la longitud eclíptica esperada. Se realizó la medición, manteniendo estático el aro eclíptico, que se mantuvo firme en esa posición todo el tiempo, encontrando el lado izquierdo del astrolabio para Capella, Proción, Marte y Arturo, tangente en las longitudes eclípticas esperadas. Después, midiendo con el lado izquierdo del aro astrolabio se encontró Regulus (Leo), en 151°, siendo la longitud prevista, 150,18°, y el error, por tanto, inferior a un grado. Para Spica (Virgo), la medición dio longitud eclíptica de 204°, siendo la prevista de 204,2°.

La conclusión que se obtiene de las mediciones realizadas, aparte de afinar los giros, es que hay que lastrar, con setecientos gramos de plomo, la parte inferior del pedestal, para estabilizar el instrumento, y evitar que el viento lo arrastre.

8. Aclaraciones sobre el nombre de los instrumentos. Astrolabio y Meteoroscopio

La **esfera armilar eclíptica**, instrumento de medición celeste, es llamada **Astrolabio** por Claudio Ptolomeo en la Composición matemática - Almagesto, Libro quinto, Capítulo primero (traducción de Halma) [7]. Transcribo literalmente el pasaje completo del mismo, traducido al español:

Libro V, Capítulo I Construcción del **astrolabio** (Almagesto)

*Tomando dos círculos bien formados alrededor, con cuatro caras perpendiculares, de las mismas proporciones en su tamaño, perfectamente iguales y parecidos entre ellos, los arreglamos de una manera que se cruzan en ángulo recto con un diámetro común. Uno representa la eclíptica, y el otro el meridiano que pasa por los polos de la eclíptica y por aquellos del ecuador. En este meridiano, tomando con el lado del cuadrado inscrito, los puntos que fijan los polos de la eclíptica; y poniendo en estos puntos, unos cilindros que salen afuera y adentro, por aquellos de afuera pasamos otro círculo cuya concavidad se adapta perfectamente a la curvatura convexa de los dos círculos en el mismo encerrado, y que puede moverse en la dirección de la longitud, girando sobre los polos de la eclíptica. A los cilindros del interior, también adjuntamos otro círculo cuya convexidad se abraza por la concavidad de los dos primeros, y que también gira en longitud alrededor de los mismos polos con el círculo exterior. Este círculo exterior y el que representa la eclíptica, dividida en 360 grados de circunferencia ordinaria, y cada uno de estos grados en tantas subdivisiones como ella puede recibir, hemos adaptado dentro de este círculo interno, otro círculo más pequeño, que se desliza por su borde convexo en la concavidad de este círculo interno, y que lleva dos pínulas prominentes y diametralmente colocadas, para que se pueda poner en movimiento en el plano del círculo interior hacia cualquier polo para observación de las latitudes. Todo esto así dispuesto, en el círculo que se concibe para pasar por los polos de la eclíptica, tomando después desde cada uno de los polos del zodiaco, el intervalo que se ha mostrado entre polos de la eclíptica y del ecuador, los puntos extremos de estos intervalos diametralmente también opuestos el uno al otro, los fijamos, como al principio de este tratado en un meridiano similar para las observaciones del arco del meridiano entre los trópicos, de modo que nuestro **astrolabio** siendo puesto en la misma posición que este instrumento, es decir perpendicular al plano del horizonte, y erigido siguiendo la altura del polo para la supuesta localización terrestre, y todo a la vez paralelo al plano del meridiano natural, los círculos internos podrían dar la vuelta a los polos del ecuador de oriente a occidente, de acuerdo con el primer movimiento del universo.*

*Estando el instrumento así colocado, todas las veces que el sol y la luna pudieron ser vistos al mismo tiempo por encima del horizonte, ponemos el círculo exterior en el grado donde encontramos más o menos que el sol estaba en ese instante, y hacemos girar el círculo que pasa por los polos, de manera que la intersección de los círculos siendo girada justo hacia el grado del sol, los dos círculos, a saber, el de la eclíptica y el que pasa por los polos de esta, se hacen sombra; o de manera que si fuera una estrella que viéramos, aplicando uno de los ojos en uno de los lados del círculo exterior dirigido hacia el grado en cuestión de la eclíptica, esta estrella se nos aparecería en el lado opuesto y en el mismo plano del círculo, como pegada a las superficies de los dos círculos. Entonces dirigimos el círculo interno hacia la luna, o hacia el astro, en cuestión, podríamos ver al mismo tiempo, la luna o el astro, objeto de nuestras investigaciones, por los dos pináculos del círculo más pequeño incrustado en el círculo interno. Así encontramos el lugar que el sol u otro astro ocupa en longitud sobre la eclíptica, hasta el punto de la intersección de este círculo con el círculo interior del **astrolabio** correspondiente en el punto análogo del círculo externo; y en grados desde este círculo, la distancia desde la luna o el otro astro sobre la eclíptica, ya sea hacia las osas o hacia el mediodía, como sobre el círculo exterior, por medio de la división del círculo interno del **astrolabio** y por el intervalo desde el medio del pináculo del círculo más pequeño que hacemos deslizar en este círculo interno hasta la mitad de la línea de intersección de este círculo y de la eclíptica.*

Otro instrumento de medición celeste, y terrestre, es llamado **Meteoroscopio** por Claudio Ptolomeo en el Tratado de Geografía, Libro primero, Capítulo tercero (traducción de Halma) [8]. Tiene algunos elementos comunes a la esfera armilar eclíptica. Transcribo literalmente el pasaje completo del mismo, igualmente traducido al español:

CAPÍTULO III.

Cómo por el número dado de estadios de una distancia cualquiera, en línea recta, no estando en el mismo meridiano, logramos conocer el número de estadios de la circunferencia de la tierra, y viceversa.

*AQUELLOS que nos precedieron, buscaron no sólo hacer un arco de gran círculo, de una distancia en línea recta sobre la tierra, sino también colocarla en el plano de un meridiano. Y observando con las escioteras (instrumentos para tomar las sombras) los dos puntos del cielo que respondían verticalmente encima de los dos extremos de esta distancia, concluyeron que el intervalo de estos dos puntos era un arco celeste, similar al camino recorrido entre los dos extremos bajo el mismo meridiano, porque, como hemos dicho, todas estas líneas están en un único plano, las rectas que parten desde estos extremos para terminar en los puntos verticales, reuniéndose en un punto común que es el centro de la esfera. Tanto pues el arco celeste, entre los puntos verticales, les parecía ser una parte de todo el meridiano, tanto suponían que la distancia terrestre era una parte similar de la circunferencia de la tierra: analogía que todavía se aplica al caso donde se toma por la distancia medida, no el arco del meridiano, sino el de otro gran círculo cualquiera, siempre que las alturas del polo igualmente se hayan observado bien en los dos extremos de esta distancia y que estemos seguros por medio del **meteoroscopio**, del cual hemos dado la construcción, de la posición de esta misma distancia relativamente al otro meridiano. Este instrumento sirve para hacer varias otras observaciones muy útiles, como tomar de día o de noche, las alturas del polo boreal para el lugar de la observación, la posición de la meridiana en cualquier instante que sea, y los de los caminos que se desvían de él, es decir, los ángulos que hacen en el punto vertical con ella el gran círculo descrito por el camino, para los cuales mostramos de forma parecida el arco buscado, por medio del **meteoroscopio**, así como el arco del ecuador, comprendido entre dos meridianos, y de la misma manera los arcos de los paralelos al ecuador. Mediante este método, una sola distancia en línea recta medida sobre la tierra, es suficiente para encontrar el número de estadios de toda la circunferencia terrestre. Y por ahí todavía, se conoce el número de estadios de las otras distancias, a pesar de que no estarían ni en una sola línea recta, ni en el mismo meridiano, ni bajo el mismo paralelo, siempre que tomáramos su oblicuidad exactamente, y las alturas de sus extremos. Porque, así nuevamente, por la razón del arco que subtiende la distancia en cuestión, a la circunferencia del gran círculo, se puede calcular fácilmente el número de estadios del contorno del globo terrestre, de acuerdo con el número encontrado de los de esta distancia.*

9. El catálogo de estrellas del Almagesto, un ejemplo de utilización de la Esfera armilar eclíptica

El Catálogo de estrellas del Almagesto determina la posición de 1022 estrellas por sus coordenadas celestes, longitudes y latitudes eclípticas [8]. Se agrupan en 48 figuras, o constelaciones, con referencias imaginarias entre ellas, y que se explican, por medio las mitologías romana, griega, babilónica, e incluso india, y anteriores. Para dar los nombres de las estrellas, se hace de acuerdo con la posición que ocupan las mismas en cada figura descrita. Sus coordenadas fueron obtenidas por medio de una esfera armilar eclíptica, llamada astrolabio, que se describe, “Sphæra solida fabricandis. . .” (“Construimos una esfera solida...”). Este Catálogo es explicado, de forma detallada, por el autor, en un artículo anterior, en el primer número de esta misma Revista, “El Catálogo de estrellas de Hiparco” [9]. Traemos, a modo de ejemplo, una figura, con la constelación zodiacal boreal de Leo, en proyección ortográfica simplificada, con latitudes no aumentadas (plate-carrée), y otra con el Catálogo completo, en proyección azimutal equidistante, realizada con Python/Cartopy.

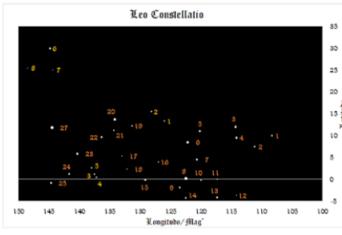
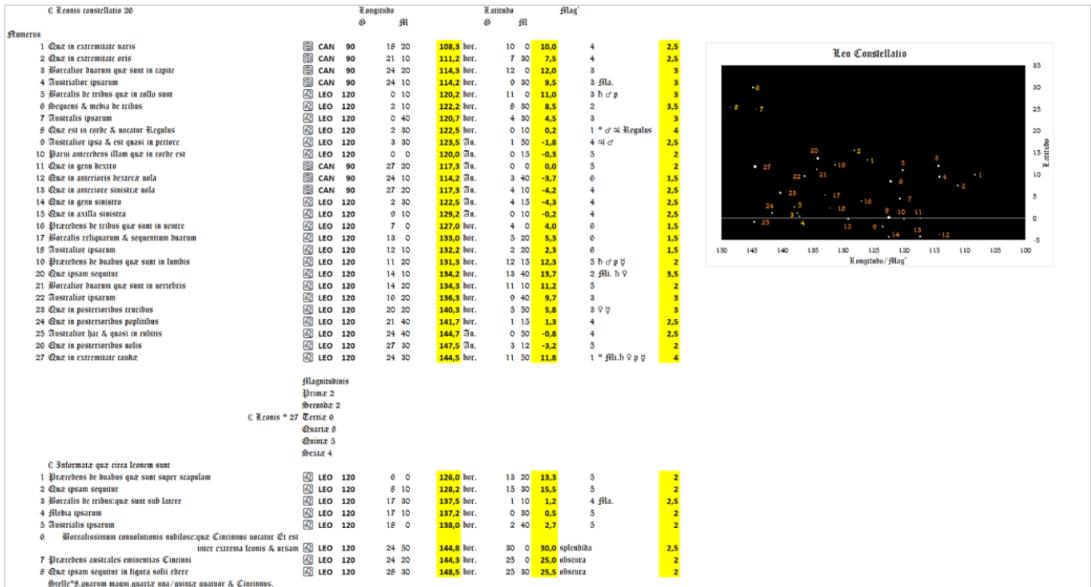


Figura 11. Constelación zodiacal boreal de Leo, obtenida del catálogo de estrellas del Almagesto.

10. Las esferas armilares a lo largo de la historia

Algunos astrónomos conocidos que usaron la esfera armilar ecuatorial fueron Kidenas (Babilonia), Eudoxo, Aristilo, Timocaris, Eratóstenes, y la esfera armilar eclíptica, Hiparco, Ptolomeo, Pappus, Teón, Proclus, Al Bateni, Al Fergani, Ibn Yunus, Azarquiel, Sacrobosco, Alfonso X, Regiomontano, Copérnico, Tycho Brahe, Johannes Kepler, . . . Nicolaus Copernicus. De Revolutionibus Orbium Caelestium (Sobre las revoluciones de las esferas celestes), Libri VI. Nuremberg, 1543 [11]. Fundamentación de la teoría heliocéntrica, que sitúa al sol en el centro del Universo. Tiene un catálogo de estrellas, con longitudes y latitudes eclípticas. Para su elaboración se utilizó la esfera armilar eclíptica. Utiliza además la regla paraláctica, también llamada triquetrum, que describe, para determinar paralajes lunares, distancias cenitales y alturas de cuerpos celestes, y está basada en el sistema de referencia horizontal. También la usaban Ptolomeo, quién también la describe, en el Almagesto, y Tycho Brahe.

11. El uso de las coordenadas celestes a lo largo del tiempo

En los tiempos más antiguos, en los que la mayoría de los astrónomos tenían una visión geocéntrica de los movimientos celestes, se usaban coordenadas celestes ecuatoriales, ascensión recta y declinación, que son geocéntricas, y se determinaban con la esfera armilar ecuatorial. A partir de la invención de la esfera armilar eclíptica, tal vez en los tiempos de Hiparco, predomina el uso de las coordenadas celestes eclípticas, también geocéntricas, que siguen el movimiento uniforme de los astros, con el fondo estelar del círculo oblicuo, reglado por el Sol, hasta el siglo dieciséis, por la época de Copérnico, en que la visión del cielo por parte de los astrónomos empieza a abandonar postulados geocéntricos, y a adoptar postulados heliocéntricos, y más allá del Sol. Sin embargo, las coordenadas más usadas, en tiempos modernos, tal vez desde la época de Tycho Brahe, o Kepler, que también usaron la esfera armilar eclíptica, y, por tanto, las coordenadas eclípticas, hasta hoy día, vuelven a ser las coordenadas celestes ecuatoriales, que no siguen un movimiento uniforme, fácilmente determinable. Tal vez tuvo algo que ver el descubrimiento del Telescopio con lentes ópticas de vidrio, que aumentan el tamaño de los objetos visuales, cuyo uso astronómico empieza con Galileo.

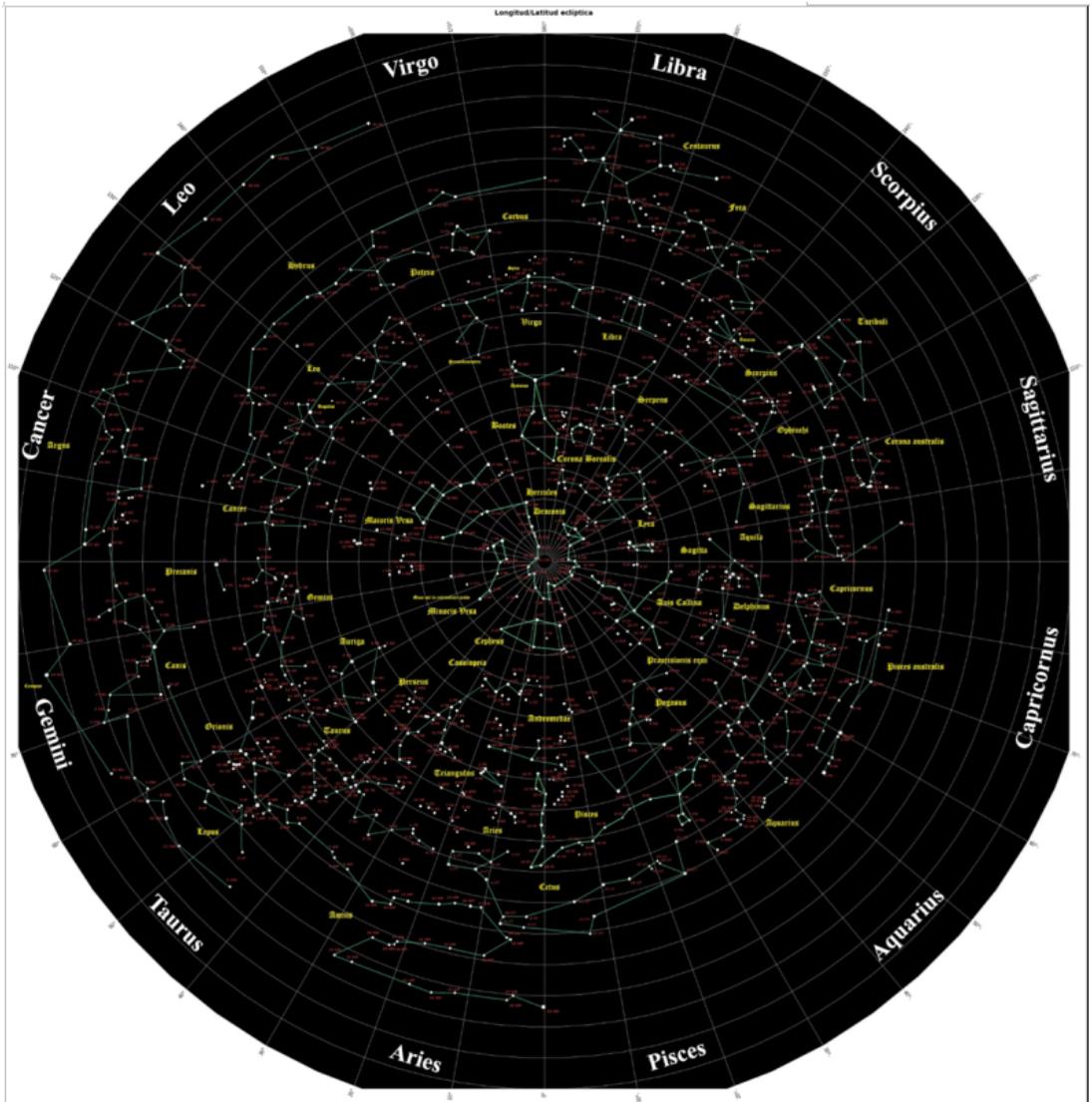


Figura 12. Catálogo de estrellas del Almagesto (Claudio Ptolomeo de Alejandría). Año primero de Antonino Pío, 138 d.C. 34 piscium en punto aries. Precesión $1^{\circ}/100$ años.

12. El Astrolabio planisférico (o plano)

Instrumento derivado de la Esfera Armilar, con la proyección estereográfica, que conserva los ángulos, pero no las distancias, y el círculo oblicuo, zodíaco, o eclíptica, con sus longitudes eclípticas. Como reloj, da horas equinocciales, según la longitud eclíptica del Sol, que se fija con la regla, usada como aguja, al círculo eclíptico, para girar solidaria con él, y la altura del astro, medida con la alidada, situada en el dorso. La longitud eclíptica del Sol se obtiene también en el dorso, en la escala de 360° , concéntrica a la escala de $365\frac{1}{4}$ días. El punto que representa al Sol se lleva al almicantrat, o círculo de alturas iguales, correspondiente, en la lámina, que depende de la latitud. También se puede, durante la noche, medir la altura de una estrella, y llevarla al almicantrat, o círculo de alturas iguales, correspondiente,

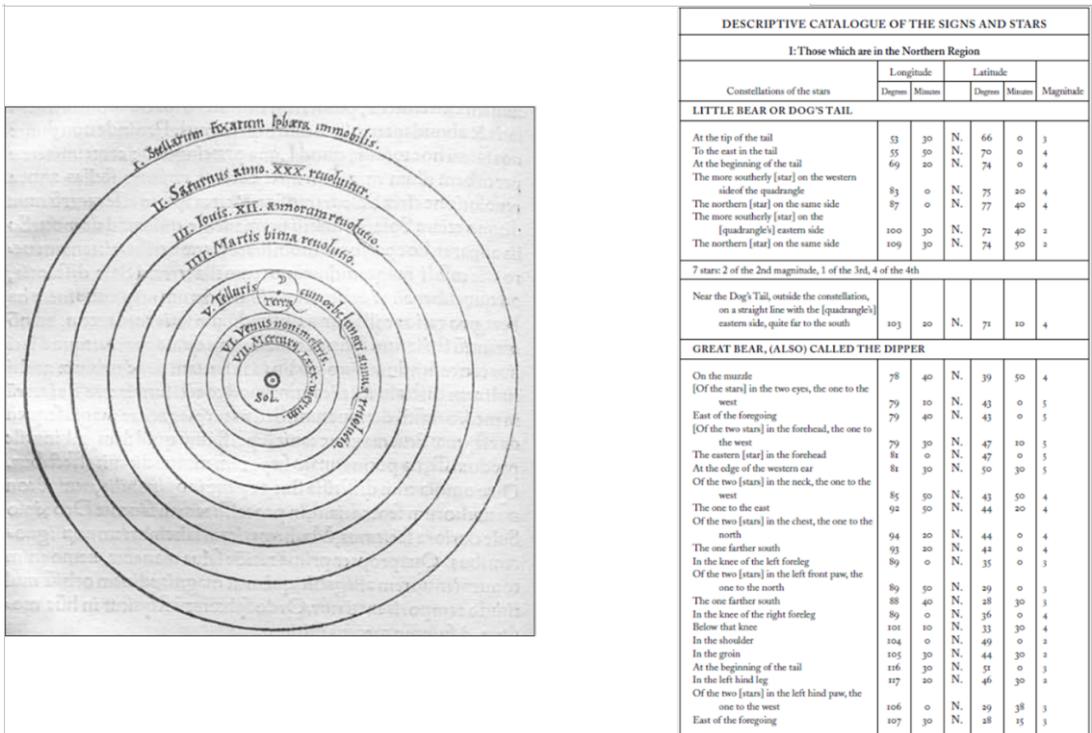


Figura 13. Nicolaus Copernicus. De Revolutionibus Orbium Caelestium.

una vez fijada la aguja a la longitud eclíptica del sol. Tiene un sistema de referencia fundamentado en el horizonte, obtenido al ser suspendido en la vertical, y se orienta de acuerdo con los puntos cardinales.

13. El paso del astrolabio esférico al astrolabio planisférico (o plano)

Es comprensible que el que orden temporal de la invención, y uso, de los instrumentos fue: primero la esfera armilar ecuatorial; después el astrolabio redondo, que era una esfera armilar portátil de pequeño tamaño; después el astrolabio plano con alidada para medir alturas; después la esfera armilar eclíptica; por último, el astrolabio plano, con proyección estereográfica, que conserva los ángulos, pero no las distancias, y es utilizable como reloj para dar las horas equinocciales (24 h/día). La proyección estereográfica se intuye en el cálculo angular que se realiza en la esfera armilar eclíptica.

14. Telescopio refractor con montura ecuatorial

Tiene un sistema de referencia fundamentado en el horizonte, que se puede nivelar con un nivel de burbuja. Una escala graduada, que se ajusta con un tornillo, permite apuntar el eje polar al polo, con la latitud correctamente ajustada, igual a la altura del polo, de forma que la ascensión recta se cambie sin más que girar una manivela, manteniendo constante la declinación. Esto contrarresta el giro horario terrestre. Estos principios geométricos son los mismos que se utilizan en las esferas armilares. Determina coordenadas celestes ecuatoriales, de la misma manera que se utilizan en la esfera armilar ecuatorial.

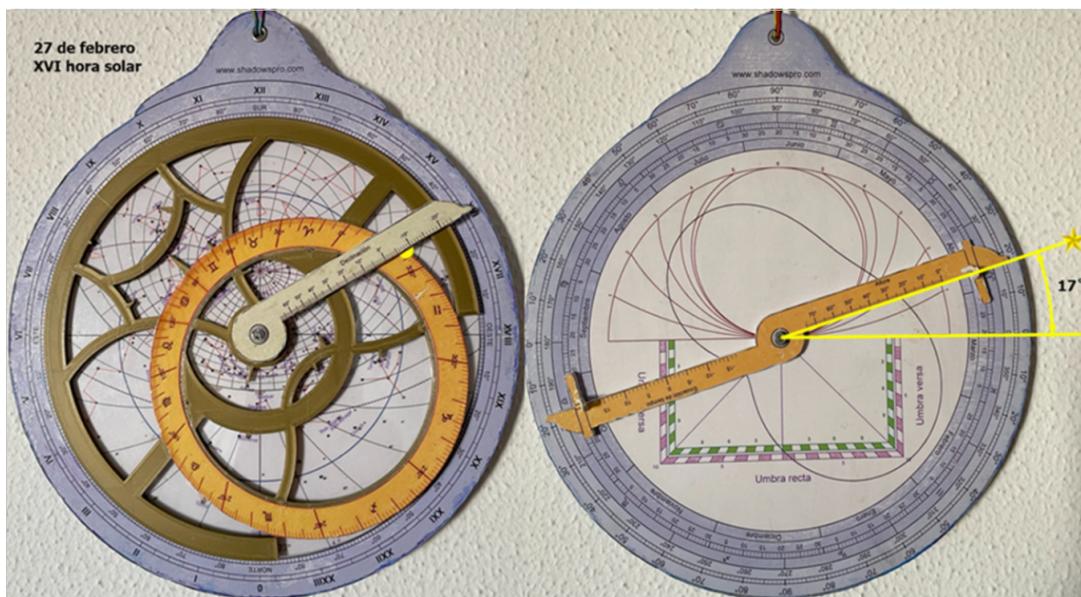


Figura 14. Astrolabio planisférico (o plano) [11].

15. Planos de referencia en los relojes de Sol

En todos los relojes de Sol, instrumentos para medir el tiempo, conocidos desde muy antiguo, y fundamentados en el movimiento aparente del Sol, encontramos sistemas de referencia y orientación, formados por planos y ejes, de las mismas características que los descritos en los instrumentos armilares: horizonte, meridiano, vertical primario o principal, ecuador, polos celestes, eje de rotación.

Reloj de sol ecuatorial: El cuadrante se pone paralelo al ecuador, de forma que el eje polar, con la aguja o gnomon, cuya sombra da la hora, apunte al polo, próximo, hoy día, a la estrella polar. También debe estar nivelado sobre el horizonte, con el cual forma el gnomon, o aguja, un ángulo llamado altura del polo, igual a la latitud. El ecuador forma con el horizonte un ángulo igual a la co-latitud (complemento a 90° de la latitud), que vale 90° en el ecuador y 0° en el polo.

Reloj de sol polar: Inclinado con respecto a la horizontal un ángulo que es igual a la co-latitud (complemento a 90° de la latitud). El eje polar apunta al polo. Los extremos del arco, llamado escocia, proyectan su sombra sobre las líneas horarias. Como en todos los relojes de sol, las sombras se alargan, y van hacia el norte, en invierno, y se acortan, y van hacia el sur, en verano, hasta los solsticios.

Reloj de sol vertical declinante: Plano vertical, y gnomon apuntado al polo, “gradus poli”, ángulo igual a la latitud.

16. Conclusión

Se ha explicado la evolución de la esfera armilar a lo largo de la historia, los distintos tipos que hay, su relación con otro instrumento astronómico parecido, llamado meteoroscopio, y su evolución hasta al astrolabio plano. Se habla acerca de los principales astrónomos que han desarrollado, y utilizado, las esferas armilares a lo largo de la historia, el tipo de coordenadas celestes que se obtienen con ellas. Se describe como se ha construido un instrumento de estas características, así como deben realizarse la preparación, y la medición de las coordenadas celestes de los astros, que permiten la elaboración de

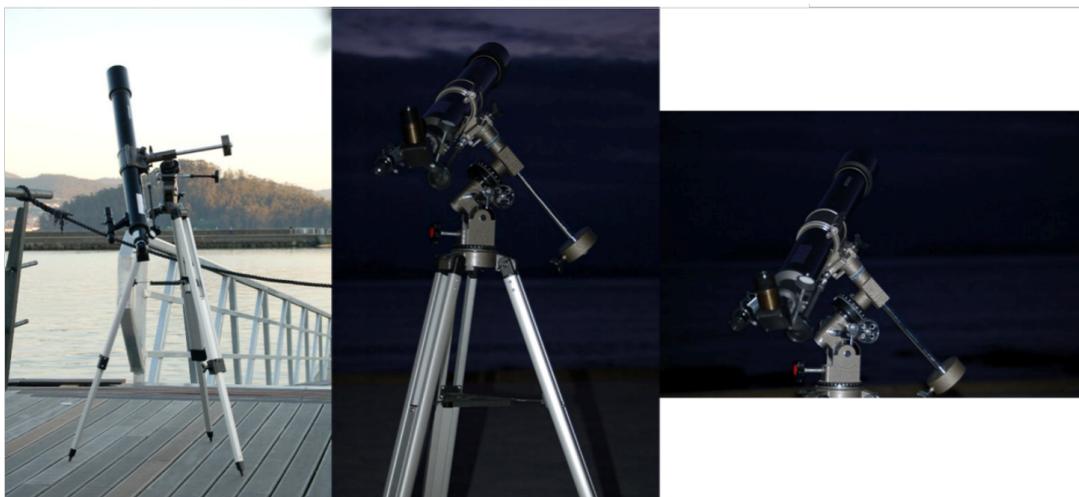


Figura 15. Telescopio refractor con montura ecuatorial.



Figura 16. Relojes de sol.

catálogos estelares. La utilización y elección de los planos de referencia, caracterizan los instrumentos, no sólo de este tipo, sino cualquier otro instrumento de medición astronómica, como los relojes de sol.

Referencias

- [1] Alfonso X de Castilla “El Sabio”. “Libros del saber de astrología” (1276). Compilados, anotados y comentados por Manuel Rico y Sinobas (1864).
- [2] J. Angelus. (1406). “Claudius Ptolemæus Cosmographia”. Latin 4804. BnF – Gallica.
- [3] F. Blayteron. (2014-2020). “Cuadrantes Solares & Astrolabios”. Manual de usuario y programa Shadows (1997-2025).
- [4] C. Ptolemæi, “Almagestum, seu Magnæ constructionis mathematicæ opus plane divinum, latina donatum lingua ab Georgio Trapezuntio, per Lucam Gauricum, recognitum anno salutis M.D.XXVIII labente”. Regiomontanus, Johannes (1436-1476). Éditeur: Luceantonii Junta (Venise) Date d’édition: 1496-1528. BnF – Gallica.

- [5] C. Ptolemæi Alexandrini. “Geographica Enarrationis, Libri octo”. Latín traducido del griego por Bilibaldi Pirckeymheri. Editor Michæle Villanuouano (1525, 1535, 1541 y 1585). BnF – Gallica.
- [6] Repositorio de código en GitHub, <https://github.com/JCAAC-FAAE>
- [7] C. Ptolemæi Alexandrini. “Geographia”. Griego y latín por Carolus Müllerus. Editor Alfredo Firmin Didot (1.883). BnF – Gallica.
- [8] N. Copernicus. “Edward Rosen translation of De Revolutionibus Orbium Coelestium”, Libri VI (Warsaw: Polish Scientific Publications, 1978), © Edward Rosen, is used with permission of Sandra Rosen.
- [9] A. K. Dambis, Yu. N. Efremov. (2000). "Dating Ptolemy's star catalogue through proper motions: the Hipparchan epoch". *Journal for the History of Astronomy*, Vol. 31, p. 115-134. May 2000.
- [10] M. A. B. Deakin, "Hypatia and her Mathematics". *The American Mathematical Monthly*, March 1994, Volume 101, Number 3, pp. 234–243.
- [11] J.-B. Joseph Delambre. (1817). “Histoire de l’astronomie ancienne”. Paris. BNF - Gallica.
- [12] J.-B. Joseph Delambre. (1819). “Histoire de l’astronomie du moyen-âge”. Paris. BNF - Gallica.
- [13] D. Re. Dicks, “Ancient astronomical instruments”, *The journal of the British Astronomical Association*, lxiv (1954), 77-85, págs. 81-83.
- [14] H. R. Dobler, "The dating of Ptolemy's Star Catalogue". *Journal for the History of Astronomy* (ISSN 0021-8286), Vol. 33, Part 3, No. 112, p. 265 - 277 (2002).
- [15] G. Gentili, Dipartimento di Matematica e Informatica “U. Dini”, Università di Firenze viale Morgagni, 67/A, 50134 Firenze, Italy. L. Simonutti, Istituto per la storia del pensiero filosofico e scientifico moderno C.N.R. Area 3 - Bicocca Milano, via Cozzi, 53, 20125 Milano, Italy. D. C. Struppa, Schmid College of Science & Technology, Chapman University One University Drive, Orange, CA 92866, USA. *Journal of Humanistic Mathematics*. Volume 10 | Issue 1 January 2020. "The Mathematics of the Astrolabe and Its History". 2020/1.
- [16] K. Gleus (Freie Universität, Berlin, Deutschland); I. Tupikova (Lohrmann Observatory, Technische Universität Dresden, Deutschland). “Astronomy and Geography Some Unexplored Connections in Ptolemy”. 2017/1/13.
- [17] O. Gingerich. (abr. 1986). “Astronomía Islámica”. *Revista Investigación y Ciencia* N.º 115.
- [18] O. Gingerich. "Was Ptolemy a fraud?". Harvard Smithsonian Center for Astrophysics, Cambridge, Massachusetts, 02138, USA (1980).
- [19] C. González Crespán (2023). “La constelación de la osa menor en la época romana (El cielo en tiempos de los romanos)”. *Asociación Astronómica de Vigo*. 2016.
- [20] C. González Crespán (2023). “La Geographia Geométrica de Claudio Ptolomeo de Alejandría”. XXV Congreso Estatal de Astronomía (FAAE), Zaragoza, 28-30 de abril de 2023.
- [21] C. González Crespán (2024). “El catálogo de estrellas de Hiparco”. XXVI Congreso Estatal de Astronomía (FAAE), Aranjuez, 31 octubre - 3 de noviembre de 2024.
- [22] C. González Crespán (2024). “El catálogo de estrellas de Hiparco”. *Journal of Computational Astronomy and Astronomical Computing (JCAAC)*, N° 1, noviembre 2024.
- [23] C. González Crespán, I. Menéndez-Pidal de Navascués (2025). “La Geografía Geométrica de Claudio Ptolomeo”. *Revista El Nuevo Miliario*. Número 20 / 21, marzo de 2025.
- [24] V. Gysembergh, A. Jones, E. Zingg, P. Cotte, S. Apicella, “Recuperado el tratado de Ptolomeo sobre el meteoroscopio”. *Archive for History of Exact Sciences* (2023).
- [25] V. Gysembergh, CNRS, UMR 8061, Francia; P. J. Williams, Tyndale House, Reino Unido; E. Zingg, Universidad de la Sorbona, Francia. “Nueva evidencia para el catálogo de estrellas de Hiparco revelada por imágenes multiespectrales” (2022).
- [26] N. Halma, “Traité de Géographie de Claude Ptolémée d’Alexandrie” (1828). BnF – Gallica.
- [27] N. Halma, “Composition mathématique de Claude Ptolémée. Tome 1 / Tome 2, traduite pour la première fois du grec en français sur les manuscrits originaux de la Bibliothèque impériale de Paris, par M. Halma et suivie des notes de M. Delambre, . . . “– 1813-1816. BnF – Gallica.

- [28] N. Halma, Théon d'Alexandrie (335? - 405?). *Commentaire de Théon d'Alexandrie sur le livre III de l'"Almageste" de Ptolémée. (Suivi des) Tables manuelles des mouvemens des astres (Nouv. éd...)* [Claude Ptolémée]; trad... du grec... par M. l'abbé Halma, . . . 1822-1825. BnF – Gallica.
- [29] A. Hernández Pérez. (2017). "Arte y ciencia en al-andalus: el astrolabio nazarí de Alcalá la Real". Departamento de Historia del Arte I (Medieval). Universidad Complutense de Madrid.
- [30] M.J.T. Lewis, "Surveying instruments of Greece and Rome". University of Hull. Cambridge University Press. (2001).
- [31] L. A. Llamazares Fernández. (abr. 2020). "El Astrolabio".
- [32] J. Martínek & A. Létal, "Astronomically determined localities, the core part of Ptolemy's Geography". *Journal of Maps*, 19:1. 2195563, DOI: 10.1080/17445647.2023.2195563
- [33] R. R. Newton, "The strange case of Claudius Ptolemy". Volume 16, Number 2. (1973).
- [34] R. R. Newton, "The crime of Claudius Ptolemy". Baltimore. (1977).
- [35] R. R. Newton, "Comments on Was Ptolemy a fraud by Owen Gingerich". (1980).
- [36] J. M. Ponce de León Álvarez. (1999). "El Astrolabio, un instrumento astronómico de las Edades Media y Moderna". *Revista de Estudios Provinciais. Excm. Diputación Provincial de Pontevedra. N.º 14.*
- [37] D. Rawlins, "An investigation of the ancient star catalog", *Publication of the Astronomical Society of the Pacific*. 94:359-373, April 1982.
- [38] A. Rome, "L'Astrolabe et le météoroscope d'après le commentaire de Pappus sur le 5º livre de l'Almageste", *Annales de la Societé Scientifique de Bruxelles, sér. A, Sci. math. Mémoires*, xlvii (1927), 77-102, págs. 77-90.
- [39] N. Sidoli, J. L. Berggren (2007). "The Arabic version of Ptolemy's Planisphere or Flattening the Surface of the Sphere: Text, Translation, Commentary". Department of Mathematics Simon Fraser University.
- [40] A. Surroca. (2019). "Azarquiel, el astrolabio y la azafea. su aportación a la ciencia astronómica y a la navegación del renacimiento". *Real Sociedad Geográfica* (2019). *Revista de Historia Naval* Núm. 144, págs. 9-34.
- [41] N. M. Swerdlow, "The Enigma of Ptolemy's Catalogue of Stars". *Journal for the History of Astronomy*, Vol.23, NO. 3/AUG, P.173, August 1992.
- [42] G. J. Toomer. (1984). "Ptolemy's Almagest Translated and Annotated by". Duckworth, London.
- [43] Mª Isabel Vicente Maroto. (Agosto 1993). "Instrumentos matemáticos del siglo XVI". *Revista Investigación y Ciencia* n.º 207.
- [44] J. Włodarczyk, "Observing with the armillary astrolabe". Polish Academy of Science. (1987).

APPLICATIONS

Estimación de los contactos C_1 y C_4 durante un eclipse solar parcial mediante análisis geométrico de vídeo astronómico

Javier de Elías¹¹Agrupación Astronómica de Madrid, Spain. E-mail: javierdeelias@gmail.com.**Keywords:** Eclipse solar, contactos

© Este artículo está protegido bajo una licencia [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/)
Este artículo adjunta un software accesible en <https://github.com/JCAAC-FAAE>

Resumen

Se presenta un método automatizado para estimar los contactos C_1 y C_4 durante un eclipse solar parcial a partir del análisis geométrico de vídeo astronómico en formato `.ser`, con timestamps individuales por fotograma. El procedimiento consiste en detectar los bordes de los discos solar y lunar mediante filtrado de bordes, ajustar sus contornos por circunferencias, y calcular la intersección entre ambas figuras. La evolución temporal del ancho de esa intersección permite extrapolar el instante en que desaparece, identificando así el contacto. El método ha sido aplicado al eclipse parcial del 29 de marzo de 2025 observado desde Majadahonda, Madrid, obteniéndose una estimación para el contacto C_1 de las 09 : 48 : 18.461 UTC \pm 2.60 s y para el contacto C_4 de las 11 : 33 : 58.787 UTC \pm 2.0 s. En el caso de C_1 se obtiene coincidencia con las efemérides dentro del rango de indeterminación, mientras que en C_4 se observa un retraso de aproximadamente 20 segundos. El enfoque propuesto es replicable por observadores amateurs con instrumental adecuado y precisión de fracción de segundo.

Abstract

An automated method is presented to estimate contacts C_1 and C_4 during a partial solar eclipse based on the geometric analysis of astronomical video in `.ser` format, with individual timestamps for each frame. The procedure involves detecting the edges of the solar and lunar discs using edge filtering, fitting their contours with circles, and calculating the intersection between the two shapes. The temporal evolution of the width of this intersection allows extrapolation of the moment it disappears, thus identifying the contact. The method was applied to the partial eclipse of 29 March 2025, observed from Majadahonda, Madrid, yielding an estimated time for contact C_1 of 09 : 48 : 18.461 UTC \pm 2.60 s and for contact C_4 of 11 : 33 : 58.787 UTC \pm 2.0 s. For C_1 , the result matches the ephemerides within the margin of uncertainty, while for C_4 , a delay of approximately 20 seconds is observed. The proposed approach can be used by amateur observers with suitable equipment and subsecond precision.

1. Introducción

La observación de eclipses solares ha sido tradicionalmente una herramienta fundamental para la validación de modelos de mecánica celeste, la mejora de efemérides astronómicas y la investigación de fenómenos solares y lunares [1]. Uno de los aspectos clave en el análisis detallado de un eclipse es la determinación precisa de los contactos, especialmente los primeros y últimos (C_1 y C_4), que marcan el comienzo y el final del fenómeno observable desde un punto determinado de la superficie terrestre. Estas definiciones se basan en la geometría estándar de los eclipses solares, como se recoge en los catálogos de efemérides de Espenak (2006) [2].

En el caso de los eclipses totales, la identificación de estos contactos puede apoyarse en señales visuales

distintivas como las perlas de Baily, la cromosfera o el anillo de diamante, lo que facilita su estimación directa mediante observación o análisis de imágenes. En contraste, los eclipses parciales presentan un reto mayor: la Luna cubre solo una porción del disco solar, y los momentos de primer y último contacto no se acompañan de fenómenos visuales fácilmente identificables. Detectar cuándo comienza o termina la ocultación resulta especialmente difícil si la fracción de disco afectada es muy pequeña, y si además la calidad óptica o las condiciones atmosféricas no son ideales.

A pesar de estas dificultades, los eclipses parciales ofrecen una oportunidad valiosa para la observación desde regiones amplias del planeta y permiten a astrónomos profesionales y amateurs participar en la obtención de datos útiles. Para ello, es fundamental contar con métodos que permitan extraer información precisa a partir de registros digitales, especialmente vídeos con marca temporal por fotograma.

En este trabajo se presenta un enfoque geométrico automatizado para la estimación de los contactos C_1 y C_4 en eclipses parciales, a partir del análisis de vídeos astronómicos grabados en formato `.ser`. Se combinan técnicas de visión por computadora para la detección de bordes y ajuste de circunferencias con análisis temporal de la posición relativa entre los discos solar y lunar. La implementación se ha realizado en Python, aprovechando librerías de código abierto, y los resultados obtenidos se comparan con predicciones teóricas para evaluar su precisión. El método se basa en la geometría de los discos lunar y solar, observados en vídeo, para estimar el instante en que la distancia entre sus centros iguala la suma de sus radios (es decir, cuando hay contacto tangencial). Se implementa el método en Python, utilizando librerías como OpenCV, NumPy y SciPy, y se valida con datos reales.

2. Metodología

2.1. Adquisición de datos

Durante el eclipse parcial de Sol del 29 de marzo de 2025 se llevaron a cabo observaciones con el objetivo específico de registrar los instantes de contacto C_1 y C_4 mediante grabación en vídeo. La adquisición de datos se realizó utilizando un telescopio Schmidt-Cassegrain de 8 pulgadas, modelo Celestron C8, equipado con un filtro solar de densidad neutra tipo ND5, adecuado para la observación segura del Sol en luz visible.

El sistema de captura incluyó una cámara astronómica monocroma ZWO ASI1600MM, conectada a un ordenador portátil que ejecutaba el programa SharpCap, configurado para grabación en modo de 8 bits. El formato elegido fue `.ser`, el cual permite la adquisición de secuencias de imágenes sin compresión y con inclusión de marca temporal individual por fotograma. La tasa de captura fue de aproximadamente 40 fotogramas por segundo.

Se realizaron dos sesiones de grabación: la primera comenzó aproximadamente un minuto antes del instante previsto para el primer contacto (C_1) y tuvo una duración de seis minutos; la segunda se inició cinco minutos antes del último contacto previsto (C_4), también con una duración de seis minutos. Esta planificación permitió cubrir con margen los cinco primeros y cinco últimos minutos del eclipse desde el lugar de observación, con el fin de obtener datos suficientes para una estimación precisa de ambos contactos.

2.2. Formato de grabación en video

El formato `.ser` es ampliamente utilizado en astronomía amateur para la grabación de vídeos astronómicos sin compresión. Fue diseñado originalmente para capturas planetarias y solares, y ofrece varias ventajas frente a otros formatos como AVI o MP4, especialmente en lo que respecta a la preservación de la calidad de imagen y la precisión temporal.

Una de las características más relevantes del formato `.ser` frente a otros formatos de vídeo convencionales es la posibilidad de almacenar una marca temporal precisa e independiente para cada fotograma. Esta funcionalidad es fundamental para aplicaciones científicas que requieren sincronización temporal exacta, como la medición de ocultaciones, tránsitos o contactos en eclipses solares. En formatos como `.avi` o `.mp4`, esta información no está presente o no se almacena con la suficiente precisión para su uso en análisis astronómicos detallados.

Un archivo `.ser` consta de tres secciones principales: un encabezado fijo de 178 bytes que almacena metadatos sobre la observación (resolución de imagen, número de frames, profundidad de bits, tipo de color, etc.), seguido por la secuencia de imágenes en bruto (una tras otra, sin compresión), y finalmente una tabla de marcas temporales que proporciona un timestamp independiente para cada fotograma.

Los timestamps se almacenan como enteros de 64 bits, representando el número de intervalos de 100 nanosegundos transcurridos desde el 1 de enero del año 1 (formato `FILETIME`, utilizado por Windows). Esta estructura permite reconstruir con gran precisión el instante de adquisición de cada imagen.

El formato permite distintos modos de codificación de imagen, incluyendo monocromo y Bayer, con profundidades de 8 o 16 bits por píxel. En este trabajo se empleó el modo monocromo de 8 bits para maximizar la tasa de fotogramas manteniendo una calidad suficiente para el análisis geométrico de los bordes del disco solar y lunar.

En los vídeos capturados, se trabajó un tamaño de imagen de 1760×2320 píxeles, y se capturaron 13187 frames para el comienzo del eclipse y 11348 frames para el fin de eclipse.

2.3. Extracción de datos del archivo

Para acceder a los datos del vídeo se utiliza una clase personalizada (`reader`) que permite interpretar correctamente la estructura interna del archivo `.ser`. En primer lugar, se lee el encabezado, que ocupa los primeros 178 bytes del archivo. Este bloque contiene información fundamental sobre la configuración de captura del vídeo, incluyendo:

- La resolución de cada fotograma (ancho y alto en píxeles),
- La profundidad de bits por canal (8 o 16 bits),
- El número total de frames del vídeo,
- El número de canales por píxel (uno para imágenes en blanco y negro o Bayer, tres para RGB),
- El tipo de codificación de color.

Con estos datos se calcula el tamaño exacto de cada fotograma en bytes. Esta información es necesaria para determinar correctamente la posición de cada imagen dentro del archivo binario y para saber dónde comienza la sección de marcas temporales.

2.4. Extracción de los timestamps

Una vez conocidos el número total de fotogramas (`n_frames`) y el tamaño de cada uno (`frame_size`), se calcula el desplazamiento hasta el bloque de timestamps como:

$$\text{timestamp_start_offset} = 178 + n_{\text{frames}} \times \text{frame_size}$$

A partir de esta posición, el archivo contiene una secuencia de marcas temporales, una por cada frame, cada una codificada como un entero de 64 bits (8 bytes) en formato *little-endian*. Cada valor representa un número de “ticks” de 100 nanosegundos desde el 1 de enero del año 1, siguiendo la convención del tipo `FILETIME` utilizado por sistemas Windows.

Para convertir cada uno de estos valores en un instante UTC real se utiliza la siguiente expresión en Python:

```
dt = datetime(1, 1, 1) + timedelta(seconds=ticks / 10_000_000)
```

Este procedimiento permite asignar a cada imagen del vídeo un timestamp con una resolución de 0.1 microsegundos, lo cual es esencial para realizar una medición temporal precisa del instante de contacto durante el eclipse.

3. Descripción del algoritmo

El algoritmo tiene como objetivo analizar cada imagen del vídeo astronómico para detectar la posición de los discos solar y lunar y calcular, en cada instante, la *profundidad de intersección*, que permitirá determinar el momento exacto del contacto (C_1 o C_4) por extrapolación. A partir de los valores obtenidos para dicha profundidad en función del tiempo, se realiza un ajuste lineal para estimar el instante en que esta se anula, lo que corresponde al contacto tangencial entre ambos discos.

El procesamiento se realiza de manera secuencial sobre los fotogramas del archivo `.ser`, con una dirección temporal que depende del contacto que se desea estudiar:

- En el caso del contacto C_1 , el análisis comienza en el último fotograma del vídeo y avanza hacia atrás en el tiempo.
- En el caso del contacto C_4 , el análisis comienza en el primer fotograma y avanza hacia adelante.

En ambos casos se parte de una geometría donde la profundidad de intersección es máxima dentro del segmento temporal capturado, lo cual permite contar con suficiente información geométrica desde el inicio del análisis.

El algoritmo se compone de las siguientes etapas:

1. Detección de bordes solares.

La imagen se convierte a escala de grises (si no lo está ya), y se aplica el filtro de Canny para detectar los bordes más marcados. Para evitar regiones contaminadas por elementos externos (como manchas solares o marca temporal), se aplica una máscara de exclusión fija allí donde es necesario. Además, se selecciona sólo la mitad izquierda o derecha del campo visual, dependiendo del contacto (izquierda para C_1 , derecha para C_4), de modo que se trabaje con una región donde el limbo solar permanece visible sin estar ocultado por la Luna.

2. Ajuste de la circunferencia solar.

Una vez detectados los puntos de borde solar, se eligen tres puntos bien distribuidos sobre el arco visible: típicamente el primero, el central y el último del conjunto ordenado. A partir de esos tres puntos se calcula una circunferencia única que los contiene. Este método evita depender de zonas parcialmente ocultadas por la Luna. El radio solar se guarda durante los primeros 30 frames para establecer una media de referencia; posteriormente se aplican criterios de estabilidad: si el nuevo radio difiere más de un 10% respecto a la media, el frame es descartado o los puntos se rotan para reintentar el ajuste.

3. Detección del borde lunar.

Una vez determinada la geometría solar, se construye una máscara circular concéntrica pero de radio ligeramente menor, que permite aislar los bordes internos al disco solar. Dentro de esta zona se buscan bordes pertenecientes al limbo lunar, especialmente en la parte superior del fotograma, donde se espera su aparición. Si se detectan al menos tres puntos adecuados, se seleccionan tres bien espaciados (nuevamente el primero, el central y el último) y se ajusta una circunferencia a dichos puntos.

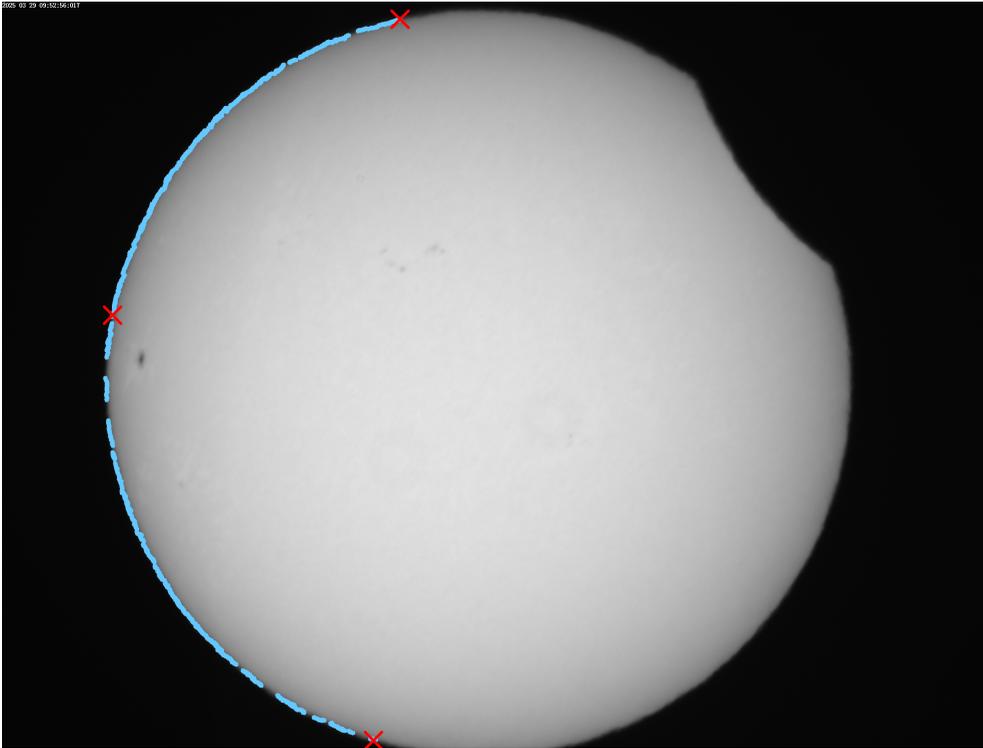


Figura 1. Detección del borde solar (en azul) y determinación de puntos (en rojo). Para la detección del borde se fuerza la mitad izquierda donde no hay eclipse..

4. Ajuste del contorno lunar.

A diferencia del ajuste solar, en el caso de la Luna se fuerza el radio a 1.05 veces el solar previamente calculado, de acuerdo con la estimación de ratio Luna/Sol en eclipse analizado. Este criterio estabiliza el resultado y reduce el error asociado a determinar una circunferencia lunar con puntos muy próximos. El ajuste consiste en encontrar el centro de la circunferencia que mejor se aproxima a pasar por los tres puntos seleccionados, manteniendo fijo el radio.

5. Intersección de discos y cálculo geométrico.

A partir de las circunferencias halladas, se calcula la distancia entre centros y se determina la *profundidad de intersección* como:

$$\text{profundidad_intersección} = R_{\odot} + R_{\text{Luna}} - d_{\text{centros}}$$

Este valor será positivo mientras exista superposición parcial entre los discos, y se anula en el instante del contacto tangencial.

6. Validación y registro.

Si el ajuste de ambos discos y la intersección son válidos, se registran los siguientes parámetros en un archivo CSV: índice de fotograma, timestamp UTC, longitud de la cuerda entre los puntos de intersección, distancia entre centros y la profundidad de intersección. Estos datos permiten posteriormente ajustar una recta a la evolución temporal de la profundidad y estimar por extrapolación el instante de contacto.

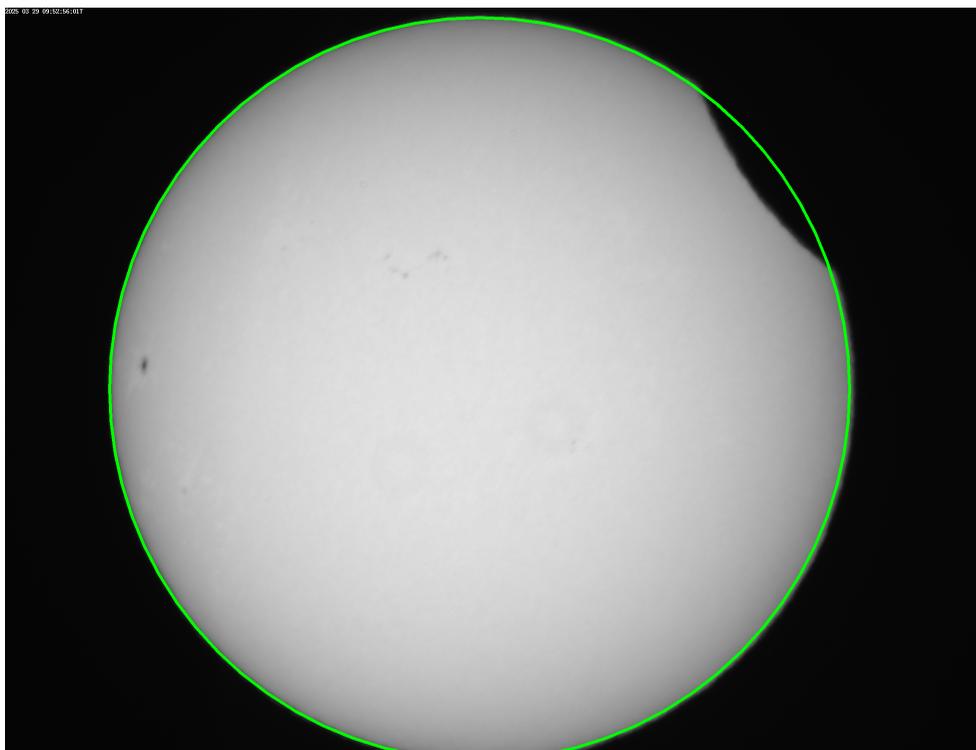


Figura 2. Cálculo de la circunferencia solar a partir de los puntos seleccionados.

3.1. Control de errores y reintentos

Dado que las condiciones de imagen pueden variar de forma significativa entre fotogramas — debido a ruido, mala definición de bordes o regiones parcialmente ocultas — el algoritmo incorpora mecanismos de control de errores y reintentos que permiten mantener la estabilidad del ajuste y evitar falsas detecciones.

En primer lugar, cada etapa crítica del procesamiento incluye criterios de validación. Por ejemplo:

- Si el número de puntos detectados en el borde solar o lunar es inferior a tres, el fotograma es descartado.
- Si el radio solar calculado difiere en más de un 10% del valor medio calculado en los primeros 30 frames, el ajuste se considera inestable y se desecha o se reintenta con puntos desplazados.
- Si no se encuentran puntos de intersección entre las circunferencias solar y lunar, el ajuste se considera inválido.

Además, en caso de que el ajuste inicial con los tres puntos elegidos no produzca una solución geométricamente válida (por ejemplo, una circunferencia indefinida o con radio anómalo), se implementa un sistema de reintentos. Este consiste en aplicar un desplazamiento cíclico (*roll*) sobre el conjunto de puntos detectados, de modo que se reevalúe el ajuste usando diferentes combinaciones de puntos.

Se permiten hasta seis intentos por fotograma. Si todos los intentos fallan, el fotograma es marcado como no aceptado y no se incluye en los resultados. Este mecanismo permite tolerancia frente a variaciones locales y evita decisiones irreversibles basadas en un único conjunto de puntos ruidosos.

Este enfoque balancea la necesidad de robustez con la eficiencia computacional, permitiendo cubrir una amplia gama de condiciones de calidad de imagen sin comprometer la fiabilidad del ajuste final.

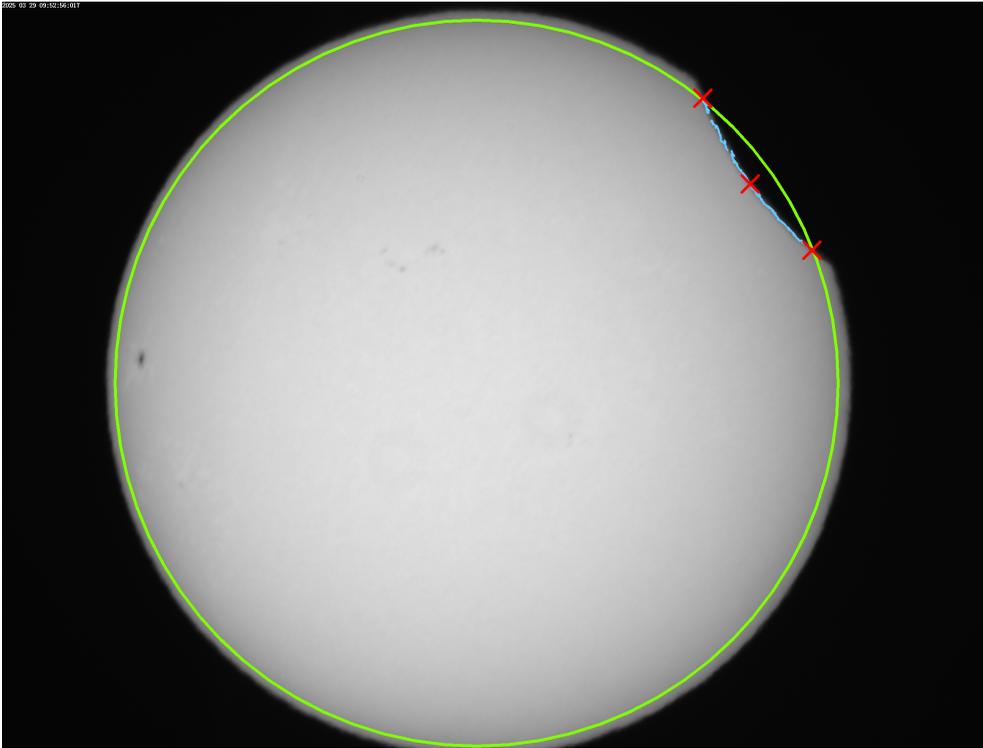


Figura 3. Detección del borde lunar (en azul) usando una máscara (contorno exterior marcado en verde) interior al círculo solar y determinación de puntos (en rojo).

4. Cálculo del instante de contacto

Una vez que se ha generado el archivo CSV con los valores de profundidad de intersección a lo largo del tiempo, se procede a estimar el instante exacto del contacto (C_1 o C_4) mediante un ajuste lineal y extrapolación. El proceso se realiza de la siguiente manera:

1. Selección del archivo y configuración temporal.

Se carga el archivo .csv correspondiente al contacto que se desea estudiar, y se define un valor de referencia: el *tiempo previsto* del contacto según efemérides.

2. Filtrado inicial.

Se eliminan del conjunto de datos todos aquellos puntos cuya profundidad de intersección supere un valor, y se restringe el análisis a los datos anteriores o posteriores a un límite temporal definido. Esta operación descarta valores en un rango cercano al contacto, que podrían deteriorar el ajuste.

3. Conversión temporal.

Se calcula la variable independiente en segundos desde el primer *timestamp* del subconjunto seleccionado:

$$t = \text{timestamp} - t_0$$

4. Ajuste preliminar y detección de *outliers*.

Se ajusta una recta por mínimos cuadrados a todos los puntos:

$$A(t) = mt + b$$

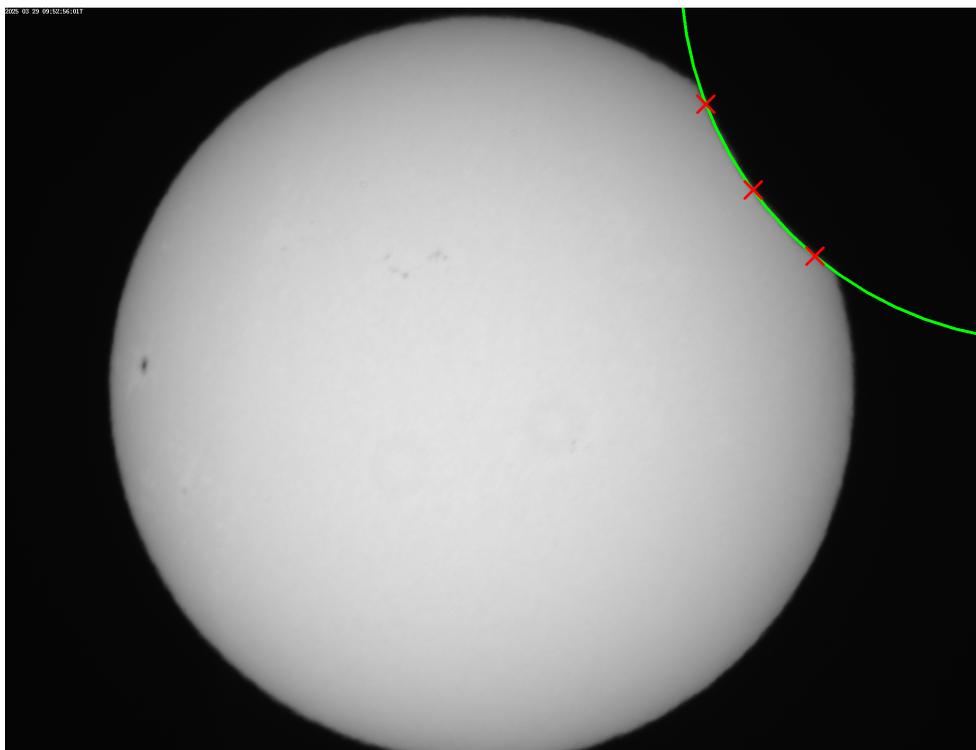


Figura 4. Cálculo de la circunferencia lunar (en verde) a partir de los puntos seleccionados (en rojo)..

Se calculan los residuos y su desviación estándar σ . Se eliminan los puntos cuyo residuo supere 2σ en valor absoluto, lo que permite depurar el conjunto de datos de valores atípicos.

5. Ajuste definitivo.

Con los datos depurados, se repite el ajuste lineal para obtener un modelo más robusto de la evolución de la profundidad de intersección con el tiempo.

6. Estimación del contacto.

El instante de contacto se estima resolviendo:

$$A(t) = 0 \Rightarrow t_c = -\frac{b}{m}$$

Finalmente, se convierte este valor en un timestamp UTC sumándolo a t_0 .

La incertidumbre asociada a este instante se ha obtenido mediante propagación de los errores estándar de la pendiente y la ordenada del ajuste lineal, aplicando los principios básicos del análisis de errores descritos por Taylor (1997) [3].

7. Comparación con el tiempo previsto.

Se calcula la diferencia entre el tiempo estimado y el tiempo previsto según efemérides, expresándola en segundos. Esto permite evaluar la precisión del método.

8. Visualización.

Se genera un gráfico con los datos originales, los datos filtrados, la recta de ajuste, y las líneas verticales correspondientes al tiempo previsto y al estimado. Esto permite inspeccionar visualmente la calidad del ajuste y la concordancia entre predicción y observación.

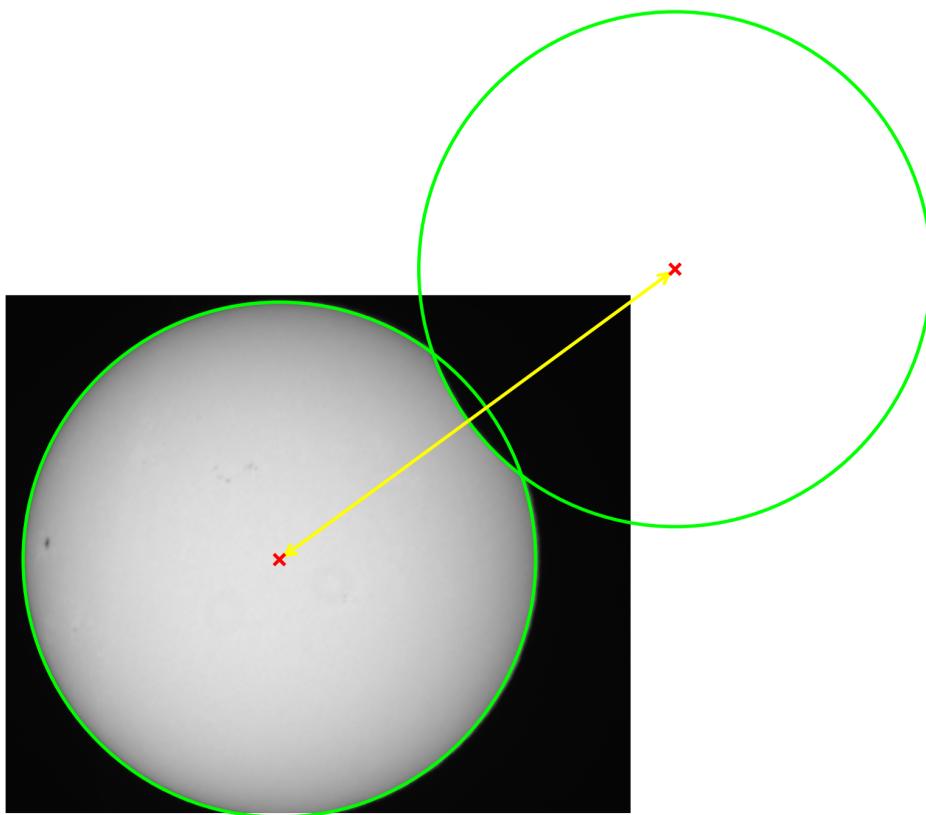


Figura 5. La profundidad de intersección se estima usando la distancia entre centros de las circunferencias calculadas.

5. Resultados

Aplicando el procedimiento descrito para la estimación de los instantes de contacto se han obtenido los siguientes resultados:

- **Contacto C₁**: Estimación temporal: 09:48:18.46 UTC \pm 2.6 s
Desviación respecto al tiempo previsto: -1.2 s
- **Contacto C₄**: Estimación temporal: 11:33:58.79 UTC \pm 2.0 s
Desviación respecto al tiempo previsto: $+20.7$ s

La desviación mostrada es en relación a lo indicado por las efemérides calculadas para la ubicación del observador (latitud $40^{\circ} 27' 38.6''$ N, longitud $3^{\circ} 52' 51.3''$ W) , según los datos proporcionados por xjubier.free.fr.

Para realizar el ajuste en cada caso, se ha aplicado un criterio de filtrado temporal para descartar medidas próximas al instante previsto de contacto, de forma que no se han considerado los últimos 90 segundos antes de C₁ y los últimos 90 segundos antes de C₄. La selección del subconjunto de datos, de acuerdo con el mencionado filtrado temporal, se ha realizado cuidadosamente mediante inspección visual previa de la nube completa de puntos. En particular, se ha puesto especial atención en conservar la tendencia lineal, evitando valores próximos al punto de cruce que no proporcionan una medida fiable

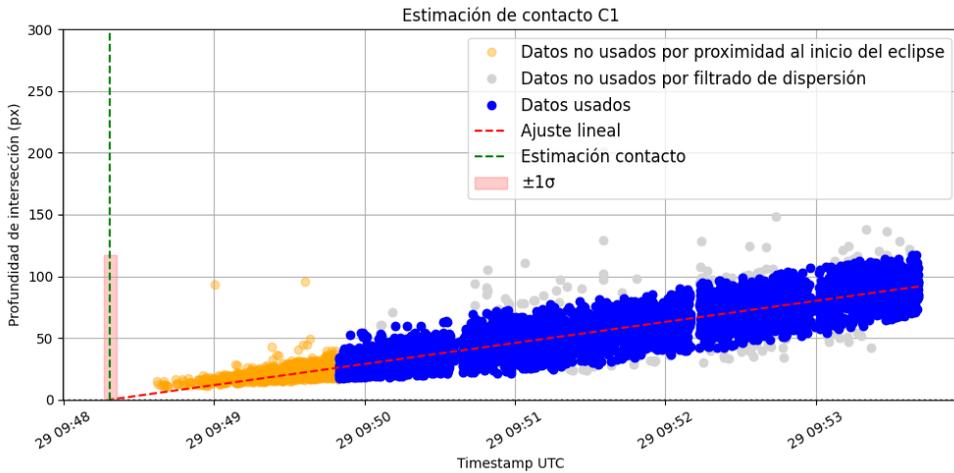


Figura 6. Estimación de C_1 .

con el presente método. Esta filtración discrecional evita que la pérdida local de linealidad comprometa la estimación temporal.

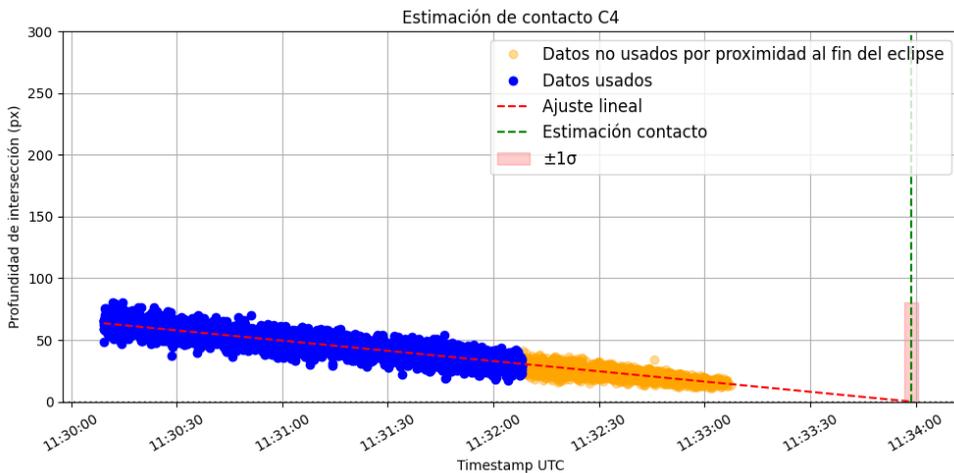


Figura 7. Estimación de C_4 .

Cabe destacar que las marcas temporales asociadas a cada fotograma son altamente fiables: se utilizó un servidor de tiempos de stratum 1 sincronizado por GPS, que emplea señal PPS (*pulse per second*), lo cual garantiza una precisión del orden de microsegundos en la marca UTC de cada imagen.

La incertidumbre en la estimación del instante de contacto se ha calculado a partir de la dispersión de los datos empleados en el ajuste lineal. Para ello, se han tenido en cuenta las incertidumbres individuales de la pendiente y la ordenada de la recta obtenida, y se ha propagado su efecto sobre la estimación final del instante en el que la intersección entre los discos desaparece. Este método proporciona una medida objetiva de la precisión del resultado, basada en la calidad del ajuste y no en supuestos externos.

6. Discusión

Los resultados obtenidos muestran, en relación con las efemérides astronómicas, coincidencia dentro de la incertidumbre del método en el inicio del eclipse (contacto C_1) pero desviación de aproximadamente 20 segundos de retraso en la terminación del eclipse (contacto C_4). Dado que se descarta un error en la sincronización temporal de los datos, esta desviación podría deberse a errores en la detección de bordes, o incluso diferencias entre el modelo astronómico de referencia utilizado por el generador de efemérides y la realidad observacional. Este comportamiento merece ser investigado en futuras campañas mediante la comparación con otros métodos independientes o con simulaciones geométricas más detalladas.

Además, el criterio de selección temporal empleado —discrecional y basado en excluir zonas con geometría más compleja— puede introducir sesgos difíciles de cuantificar sin un análisis de sensibilidad más riguroso. Aun siendo razonable en su motivación, este procedimiento introduce una incertidumbre no explícita en la estimación, que debe tenerse en cuenta al interpretar los resultados. La incertidumbre final se ha calculado considerando la dispersión de los datos ajustados y propagando sus errores hasta el instante estimado de contacto, lo que permite expresar el resultado con un margen temporal bien definido.

La asunción de que la profundidad de intersección evoluciona de manera lineal cerca del contacto también puede ser una simplificación excesiva, especialmente si hay irregularidades en la forma de los bordes detectados o si la Luna no avanza exactamente en dirección perpendicular al limbo solar.

En futuras campañas podría explorarse aumentar el número de puntos para la detección de los radios, la sensibilidad del método a la elección de los puntos de ajuste, así como incorporar métodos de ajuste no lineal o modelos geométricos más completos. También sería útil comparar los resultados con observaciones independientes (visuales, espectroscópicas o fotométricas) o con simulaciones precisas de la geometría solar-lunar. También sería interesante comparar los resultados obtenidos con predicciones generadas mediante software especializado como *Occult 4* [5], que permite calcular con precisión los instantes esperados de contacto según modelos astronómicos completos.

Herramientas como *Solar Eclipse Timer* han sido desarrolladas precisamente para facilitar la planificación y sincronización de observaciones de contactos [4], aunque no aportan resolución suficiente para una determinación geométrica automatizada.

La observación visual de fenómenos como las Perlas de Baily o el anillo de diamante también puede ser útil para validar experimentalmente los instantes de contacto [6].

7. Conclusiones

Se ha presentado un método geométrico automatizado para la estimación de los contactos C_1 y C_4 en eclipses solares parciales, a partir del análisis de secuencias de vídeo grabadas en formato `.ser` con marcas temporales de alta precisión. El enfoque combina detección de bordes, ajuste de circunferencias y análisis temporal para inferir el instante de contacto mediante extrapolación de la profundidad de intersección entre los discos solar y lunar.

Los resultados obtenidos son coincidentes con las efemérides en el inicio, pero muestran una desviación de aproximadamente 20 segundos de retraso en la terminación del eclipse. Se han mencionado posibles fuentes de explicación de esta desviación, como la pérdida de linealidad en los datos cercanos al cruce, una menor calidad geométrica en los últimos fotogramas, o errores sistemáticos en la detección del borde lunar. Cabe señalar que las efemérides utilizadas como referencia se basan en modelos astronómicos que incluyen supuestos específicos sobre los radios aparentes del Sol y la Luna, así como sobre la definición geométrica exacta del instante de contacto. Por tanto, parte de la desviación observada podría atribuirse a diferencias entre los criterios empleados en el cálculo y los aplicados en el presente método experimental. Esta posibilidad refuerza la necesidad de comparar sistemáticamente los resultados con

efemérides generadas mediante distintos softwares y marcos de referencia.

Este trabajo pone de manifiesto la viabilidad de realizar medidas temporales precisas en eclipses parciales con equipamiento asequible y técnicas de análisis reproducibles. En futuras observaciones se propone mejorar la modelización geométrica, refinar el tratamiento de los bordes detectados, y contrastar los resultados con métodos alternativos o simulaciones detalladas para consolidar la robustez del enfoque.

References

- [1] J. M. Valenzuela and R. Valenzuela (2021). *Sizing the Sun from a Solar Eclipse*. <https://aasnova.org/2021/10/29/sizing-the-sun-from-a-solar-eclipse/>
- [2] F. Espenak (2006). *Fifty Year Canon of Solar Eclipses: 1986–2035*. NASA/TP–2006–214141. <https://eclipse.gsfc.nasa.gov>
- [3] J. R. Taylor (1997). *An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements*. 2nd edition. Sausalito, California: University Science Books.
- [4] Gordon (2023). Solar Eclipse Timer - Help File. <https://www.solarecliptimer.com/help-file.html>. Accessed: 2025-04-01.
- [5] D. Herald (2023). Occult 4 - Software for Occultation Predictions. <http://www.lunar-occultations.com/iota/occult4.htm>. International Occultation Timing Association
- [6] Eclipse Tours (2023). Eclipse Edge Observation. <https://www.eclipsetours.com/eclipse-edge-observation/>. Accessed: 2025-04-01. (IOTA).

SECTION: HERRAMIENTAS DEL OBSERVATORIO VIRTUAL

Caracterización de cúmulos abiertos con Clusterix 2.0

Joaquín Álvaro Contreras¹

¹FAAE, Madrid, Spain. E-mail: jalvaro@citelan.es.

Keywords: cúmulos estelares, herramientas VO, TOPCAT, Aladin, observatorio virtual, Gaia

© Este artículo está protegido bajo una licencia [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/)

Resumen

En este segundo artículo de la serie dedicada a herramientas del Observatorio Virtual (VO) aplicadas a la caracterización de cúmulos estelares abiertos (OCs) vamos a centrar nuestra atención en Clusterix 2.0 [1]. Clusterix 2.0 (L. Balaguer-Núñez, *et al.*) [2] es una herramienta interactiva, basada en la web y compatible con otras herramientas VO, para la determinación de probabilidades de pertenencia a cúmulos estelares basándose en datos de movimiento propio y utilizando para ello un método totalmente no parametrizado. Exploramos en este artículo el uso de Clusterix aplicándolo igualmente a casos prácticos y veremos también la interoperabilidad de Clusterix con TOPCAT [3], Aladin [4] y VOSA [5].

Abstract

In this second article of the series dedicated to Virtual Observatory (VO) tools applied to the characterization of open star clusters (OCs), we will focus our attention on Clusterix 2.0 [1]. Clusterix 2.0 (L. Balaguer-Núñez, *et al.*) [2] is an interactive, web-based tool compatible with other VO tools, designed to determine membership probabilities in star clusters based on proper motion data using a fully non-parametric method. In this article, we explore the use of Clusterix by applying it to practical cases and also discuss its interoperability with TOPCAT [3], Aladin [4] and VOSA [5].

1. Introducción

Tal como vimos en el anterior artículo de esta serie, la determinación de los componentes de un cúmulo estelar abierto (OC) es un proceso que, si bien no presenta demasiadas dificultades utilizando herramientas VO como TOPCAT [3], puede resultar en cierto modo tedioso y no falto de la sensación de una dependencia directa sobre apreciaciones subjetivas del analista, al menos en los primeros pasos donde la detección de sobre-densidades en el espacio de configuración de movimientos propios va a estar condicionada de alguna manera por la experiencia previa de éste. Incluso, como también vimos, la selección inicial marcando *a mano alzada* la posible *sobre-densidad* puede parecer un procedimiento poco ortodoxo y *artesanal*.

Para compensar esta aproximación inicial (*a ojo*) nos veíamos obligados a ir refinando el procedimiento con sucesivas etapas de filtrado, lo que supone hacer de este método un proceso inevitablemente supervisado y parametrizado.

Clusterix supera estas etapas iniciales con una determinación empírica de las funciones de frecuencia a partir del diagrama de puntos vectoriales (VPD), (movimientos propios estelares), sin depender de ninguna suposición previa sobre sus perfiles [2]. En el área ocupada por un cúmulo, la función de frecuencias está formada por dos contribuciones: las estrellas del cúmulo y las estrellas de campo. El

objetivo es obtener una separación óptima de las dos poblaciones.

El *VPD* se divide en un número grande de celdas. Cada celda se describe por su posición en la matriz total definida por filas y columnas, (i, j) , y los movimientos propios correspondientes a sus respectivos centros, (a_i, b_j) . A cada celda se asigna un valor empírico para la función de frecuencia:

$$\psi(a_i, b_j) = \sum_{k=1}^N k(a_k, b_k) \quad (1.1)$$

donde $k(a, b)$ es la llamada *kernel function* (Galadi-Enriquez, et al.) [6]:

$$k(a, b) = \frac{1}{2\pi h^2} \exp \left[-\frac{(a - a_i)^2 + (b - b_i)^2}{2h^2} \right] \quad (1.2)$$

siendo h un parámetro de *suavizado*, (*smoothing parameter*), que ajusta la propia herramienta. El tamaño de las celdas se puede elegir para un análisis rápido (*Normal*) o más refinado (*High precision*).

2. Trabajando con Clusterix 2.0

Sin más preámbulos vamos a iniciar el desarrollo de un caso práctico. Volvemos sobre el cúmulo NGC 2682 que nos permitirá comparar resultados con el trabajo analizado en el artículo anterior. Posteriormente se propondrán varios ejemplos más para ser desarrollados por el lector.

Clusterix 2.0 es una herramienta web. No es necesario por tanto instalar ninguna aplicación adicional en nuestro ordenador, (salvo las ya utilizadas anteriormente: TOPCAT y Aladin). Empezaremos por acceder a Clusterix: <http://clusterix.cab.inta-csic.es/clusterix/>. (Es conveniente asegurarse que se hace desde un navegador compatible. Testado para Chrome, Firefox, Safari, ...).

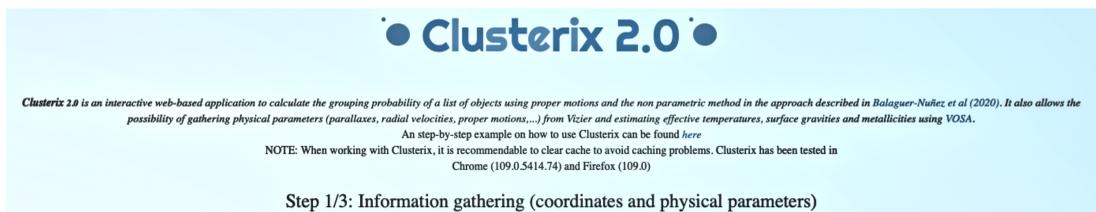


Figura 1. Carátula de presentación de Clusterix 2.0. Ya se advierte que serán tres las etapas del proceso a seguir.

La primera pantalla nos ofrece dos opciones de búsqueda (figura 2): por ID del objeto deseado, (por ejemplo, NGC 2682), o por sus coordenadas. Más tarde necesitaremos estas coordenadas, por lo que de una u otra forma conviene tenerlas a mano. Podemos recurrir para esto a un atlas del cielo (*Aladin*, por ejemplo) o al catálogo de OCs ya utilizado en el artículo previo, (*CDS Vizier* [7]).

Además de centrar una región de cielo, (por ID del objeto, o por las coordenadas), deberemos indicar el radio del cono que define el área de análisis. En este caso tomamos un valor significativamente mayor (3°) que el utilizado en el artículo anterior para NGC 2682 con TOPCAT, que fue de 1° . Igualmente se indica el catálogo desde el que se quiere recopilar la información. Seguiremos utilizando *GAIA/EDR3* [8]. El parámetro *RUWE*, marcado por defecto en un valor de *1.4*, sirve para excluir objetos extensos en el campo de trabajo, (por ejemplo, galaxias). También se puede acotar el número de estrellas que se recogerán desde *GAIA/EDR3* poniendo límites (*min, max*) en magnitud visual aparente. No tocaremos aquí y, ya, solicitamos la búsqueda.

Figura 2. Definición del campo de cielo a estudiar. Puede hacerse la búsqueda por ID del objeto, o por sus coordenadas. Aquí deben completarse igualmente algunos parámetros para recopilar la información inicial necesaria.

En pocos segundos la búsqueda nos devolverá una lista de objetos, en este caso 131 779 estrellas, y un mapa simple sobre el que se puede hacer zoom. Igualmente aparecen tres opciones: descarga de la lista generada, envío a otras herramientas VO, (por ejemplo, TOPCAT, Aladin; si previamente hemos abierto estas aplicaciones). La tercera opción 'componentes a partir de los movimientos propios', marcada con una flecha en la figura 3, nos lleva al segundo paso en el análisis con Clusterix, (figura 4).

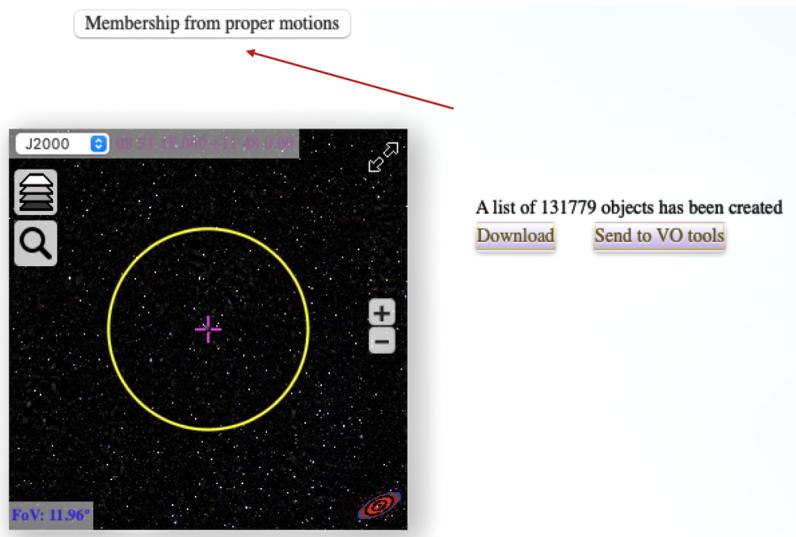


Figura 3. Datos recopilados de GAIA/EDR3 sobre la región de cielo solicitada.

A pesar de que Clusterix se auto define como herramienta no parametrizada, ('no parte de ningún supuesto sobre valores paramétricos de los componentes del cúmulo'), en esta etapa del proceso, (figura 4), debemos configurar diversos parámetros de los que depende sensiblemente el resultado esperado.

Lo primero que debemos definir son tres áreas o regiones:

1. radio del campo de visión en el que se espera estén contenidos los componentes del cúmulo. En este área habrá, por tanto, estrellas del cúmulo y otras que no lo son. [*cluster + field*]
2. un área vacía, que puede estar *contaminada* por estrellas del cúmulo y que se excluye del análisis de frecuencias.
3. y finalmente otra región donde se asume que ya no hay estrellas componentes del cúmulo. Esto, es, de sólo estrellas de campo.

Esto es algo parecido a lo que se hace en fotometría diferencial cuando se establecen *radios de apertura* sobre un objeto para definir su luminosidad comparada con otras fuentes del campo de trabajo.

Clusterix 2.0

Step 2/3: Region selection

Cluster info: 132.825,11.8_180_arcmin_GAIAEDR3

Selection of the "cluster+field" and "only field" regions

Click *Drawing Info* button if you need help on how to make the region selection

Area definition: Cluster+Field Void Field

Clear

Cluster+field:	Cluster+field area
<input type="text" value="132.825,11.800,1.0;"/>	<input type="text" value="3.141592653589793"/>
Void:	Void area
<input type="text" value="132.825,11.800,2.0;"/>	<input type="text" value="9.42477796076938"/>
Only field:	Field area
<input type="text" value="132.825,11.800,3.0;"/>	<input type="text" value="15.707963267948966"/>

Membership determination parameters

Proper motion limits (mas/yr)
Maximum μ : Maximum μ err:

Magnitude range: \leq mag. \leq

Smooth param (mas/yr) (?):

Fine tuning values
y threshold (?):
Empirical frequency function min value \rightarrow Probability min value

 \leq pmRA \leq
 \leq pmDEC \leq

Go to step 3

Matrix size ?
 Normal High precision
Total number of stars: 110629
Number of stars in the "cluster+field" region: 13989
Number of stars in the "field" region: 60181
Field sample size? 60181

Warning: The number of objects is large and the computation of the frequency functions can take a long time. You'd better reducing the sample size to a value lower than 50000

J2000

FoV: 7.83°

Drawing info

Figura 4. Paso 2. Definición de 'regiones' y otros parámetros.

La manera de definir estas tres regiones es escribiendo en cada una de las *casillas* un centro y un radio: ((AR,Dec),radio). Los valores del *centro*, iguales en los tres casos, corresponden a las coordenadas del objeto de estudio determinadas en el primer paso. Los valores de los radios son progresivamente crecientes, como es lógico, empezando por el que asumimos que contiene al cúmulo y terminando en el radio del campo del que se tienen estrellas, (el fijado también en la selección de objetos del primer paso). En los tres casos es importante!! terminar con `';`.

Los resultados obtenidos dependen críticamente de una buena definición de estas tres regiones. Si no se obtiene un resultado que pueda considerarse válido, conviene probar con otros valores para los radios elegidos. En el mapa adjunto (figura 4, derecha) se dibujan las tres regiones.

Aunque *no se asumen valores* para los diferentes parámetros de los objetos en el análisis, las dos 'viñetas' que siguen, (bajo la selección de regiones), permiten restringir algunos valores con objeto de hacer más liviano el cálculo de la función de frecuencias: por ejemplo, limitando el valor máximo de *movimientos propios*, el *error máximo*, y/o el *rango de magnitudes*. En el ejemplo desarrollado aquí acotamos a 15.0 el máximo en movimientos propios y a un 10% el valor del error relativo asumible.

El valor del parámetro de *smoothing* lo ajusta Clusterix, (como ya se dijo), aunque puede modificarse. Un valor demasiado alto desdibujará los detalles de la función de frecuencias, mientras que un valor bajo producirá un resultado con más *ruido*. Es aconsejable no modificar el valor calculado y, en cualquier caso, si fuera necesario es preferible redefinir los radios de las regiones seleccionadas.

De igual manera, Clusterix por defecto establece un valor umbral (*threshold*) de 3.0 para el cálculo de la probabilidad. Como la función de frecuencia estimada de "campo único" no es una representación perfecta de la verdadera función de frecuencia de campo, esto puede producir valores de probabilidad sin sentido. Para evitar esto, Clusterix restringe los cálculos de probabilidad a los casos donde la *relación señal-ruido* es tres veces superior a este umbral. Este umbral también es modificable y en el caso de resultados poco satisfactorios, se puede establecer en valores ligeramente más altos. Lo preferible en principio es no tocarlo.

Finalmente, en la parte derecha de esta página, (figura 4), tenemos opción a elegir el *tamaño de la matriz* $k(a_k, b_k)$: *normal* (100 x 100), *high precision* (300 x 300), y si el número de estrellas en la *región de campo* supera las 50 000, aparecerá un *warning* indicando que el cálculo puede hacerse pesado. Si así ocurre, puede reducirse este valor con el punto de scroll sobre el warning hasta dejarlo por debajo de las 50 000 estrellas de campo.

2.1. Resultados

Una vez concluido el análisis en esta etapa, (es visible una barra de progreso en la parte inferior de la página), se presenta un cuadro con 4 gráficas como el de la figura 5. (Después de cada modificación en las regiones $c+f$ y f el sistema recalcula y dibuja las funciones empíricas de frecuencia, $\psi_c = \psi_{c+f} - \psi_f$. Clusterix nos ofrece 2 304 candidatos a formar parte de NGC 2682, (del total de las 131 779 estrellas presentes en la región del cielo analizada). Lo interesante aquí es que, además de haber localizado la *sobre-densidad* en movimientos propios propia del cúmulo, se asigna un valor de *probabilidad* a cada una de las estrellas del campo sobre su pertenencia al cúmulo. Esto queda patente en la gráfica de la parte inferior-derecha de la figura 5. Incluso los pequeños grupos, menos significativos, que aparecen en dicha gráfica, podrían revelar otros posibles cúmulos no contemplados inicialmente.

Este campo de probabilidad es un simple cálculo sobre las funciones de frecuencia de las regiones 1 y 3, (los subíndices 'c' y 'f' hacen referencia a estas regiones, cúmulo y campo, respectivamente):

$$P = \frac{\bar{\psi}_c}{\bar{\psi}_c + \bar{\psi}_f} \quad (2.1)$$

Si pasamos a la etapa 3 del proceso, (botón 'Go to step 3' - ver Fig. 4), tendremos un resumen pormenorizado de estos resultados (Fig. 6). Clusterix facilita una tabla en la que, a los datos recopilados inicialmente de *GAIA/EDR3* para cada una de las 131 779 estrellas, añade tres campos más:

1. **PROB**: probabilidad que se asigna a cada estrella de ser componente del cúmulo.
2. **FLAG**: con valores $M = \text{miembro del cúmulo}$ o $NM = \text{no miembro}$.

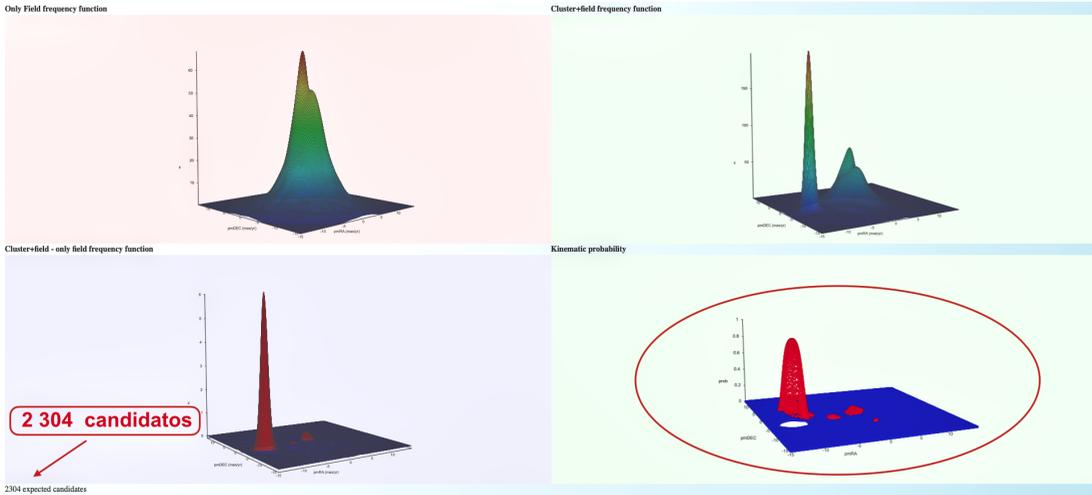


Figura 5. Paso 2. Resultados: representación de la función de frecuencias en las diferentes regiones, y del campo de probabilidad en la viñeta inferior-derecha.

3. y, **Región** en la que cada estrella se encuentra: F , C y V .

Esta tabla puede ser descargada en formato *.txt* para posterior estudio/manipulación o bien enviada a otras herramientas VO, como TOPCAT, para continuar un análisis más refinado, algo a todas luces conveniente. Para que este envío se produzca, antes de pulsar el botón 'Send to VO tools' deberemos abrir la aplicación TOPCAT.

3. De nuevo en TOPCAT

Clusterix ya nos ha ofrecido un conjunto de posibles componentes del cúmulo: estrellas identificadas y disponibles en el fichero *.txt* listo para descargar. Podemos asumir que el trabajo está completado ya en este momento. En este sentido Clusterix es autoconsistente y no requiere más apoyo, salvo que parezca necesario o conveniente repetir el análisis ensayando algún cambio en las condiciones de contorno definidas en el proceso con objeto de confirmar resultados o mejorar los mismos, si no hubieran sido demasiado convincentes.

Sin embargo, como vamos a ver, no se deben ignorar las funcionalidades de TOPCAT para testear los datos obtenidos e incluso mejorar el análisis. Por tanto resulta aconsejable *abrir* TOPCAT y proceder a transferir los datos desde Clusterix, ('Send to VO tools').

En una primera exploración de los datos transferidos, ('Display table cell data', cuarto icono por la izquierda en el conjunto de herramientas de la parte superior en la ventana de TOPCAT, Fig. 7), podemos ver que la tabla está ordenada por el valor del campo 'PROB': aparecen primero los 2 305 objetos calificados como *miembros* del cúmulo, y esto nos da información acerca del criterio seguido por Clusterix en este caso: $member \equiv PROB \gtrsim 70\%$.

También hay que resaltar que entre los candidatos a 'miembros' hay estrellas en cualquiera de las tres regiones: C , F , V . Esto es algo que se aprecia mejor en un mapa (Fig. 9).

Lo primero que haremos es crear el *sub-set* que separa *miembros* de *no-miembros*. Para ello definimos este subconjunto, al que llamaremos '**M67v1**' (siguiendo la nomenclatura del artículo anterior), creando para ello un primer filtro, Fig. 8:

$$Expression : equals(FLAG, "M") \tag{3.1}$$

Step 3/3. Determination of membership probabilities

Send to VO tools

Save as text

```
# Results were retrieved using Clusterix software
# http://clusterix.cab.inta-csic.es/
# In case of problems, please, report to: clusterix_archive_support@cab.inta-csic.es
#
# Labels:
#
# STAR_NO      identifier of star retrieved from the input data
# RAJ2000     right ascension of a star
# DECJ2000    declination of a star
# pmRA        proper motion in right ascension
# epmRA       error in proper motion in right ascension
# pmDEC       proper motion in declination
# epmDEC      error in proper motion in declination
# Pll         Parallax
# ePll        error in parallax
# BP          BP magnitude
# eBP         error in BP magnitude
# RP          RP magnitude
# eRP         error in RP magnitude
# G           G magnitude
# eG          error in G magnitude
# RV          Radial velocity
# eRV         error in radial velocity
# Source_ID   Unique Source Identifier
# RUWE        re-normalised unit weight error
# PROB        probability that star belongs to evaluated open cluster
# FLAG        M=Membership NM=No membership
#
# Region values: F = star was inside a "only field" region
#                  C = star was inside a "cluster+field" region
#                  V = star was outside the selected regions
#
# Parameters:
#
# CLUSTER INFO: 132.825,11.800_180_arcmin_GAIAEDR3
# PROPER MOTION CUTOFF: 15.00 mas/yr
# PROPER MOTION ERR CUTOFF: 10.00 mas/yr
# SMOOTH PARAMETER: 0.62
# GAMMA FACTOR: 5.00
# QUERY MIN MAG (STEP 1):
# QUERY MAX MAG (STEP 1):
# QUERY MAX RUWE (STEP 1): 1.4
# FILTERED MIN MAG (STEP 2):
# FILTERED MAX MAG (STEP 2):
```

Figura 6. Paso 3. Resumen de resultados y otras opciones.

También se podría en este punto hacer una selección más restrictiva asumiendo como miembros del cluster sólo aquellos objetos con una probabilidad asignada más alta, por ejemplo del 80 %: *Expression* : $PROB \geq 0.8$. Seguiremos aquí por ahora con el criterio original sugerido por Clusterix.

Hecho esto, podemos ver las propiedades de los componentes propuestos frente al resto de objetos en el mismo campo de cielo.

Lo reflejado en la Fig. 9 (izq.) es interesante, porque nos está diciendo que:

1. Clusterix no restringe la asignación de *membresía* al espacio definido en el campo 'C'. Sólo utiliza este área para establecer la función de frecuencia ψ_c , por lo que la elección del radio para esta región en el paso 2 de Clusterix no tiene por qué contemplar una estimación previa, más o menos rigurosa, del tamaño del cúmulo, sino que basta con considerar que sea suficiente para contener un número significativo de componentes del mismo.

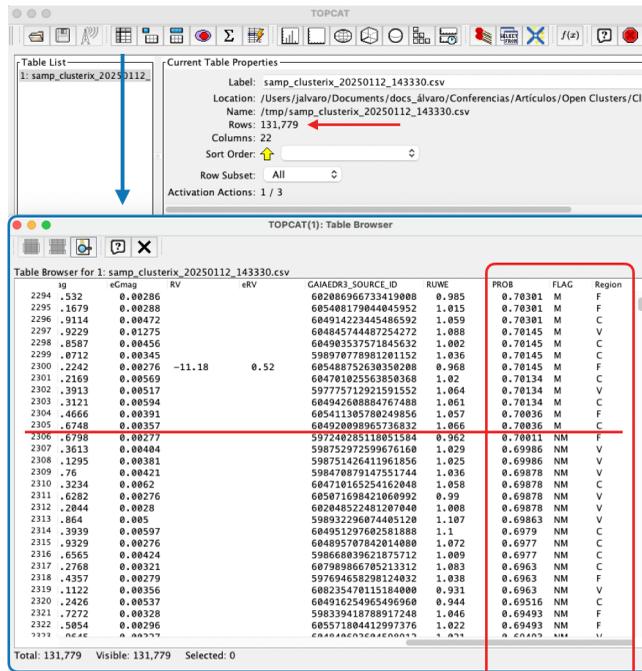


Figura 7. Tabla transferida desde Clusterix a TOPCAT. Están todos los registros analizados por Clusterix e identificados aquellos que han sido considerados como candidatos a formar parte del cúmulo.

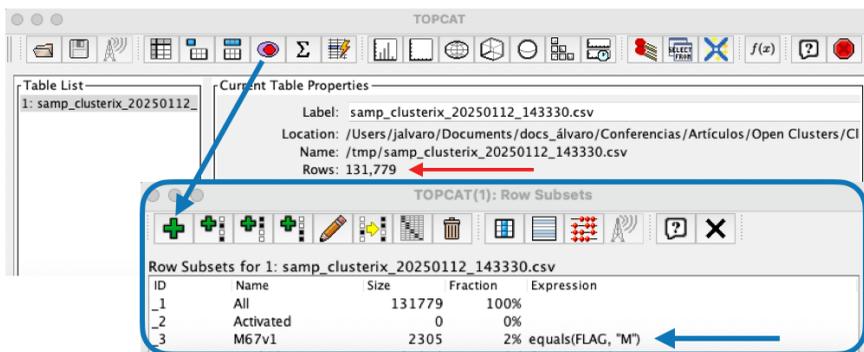


Figura 8. Nuevo sub-set, M67v1, integrado por los candidatos propuestos por Clusterix a formar parte del cúmulo: FLAG = 'M'.

2. Por otra parte, el hecho de encontrar miembros asignados incluso hasta el extremo exterior del área de campo (3º) puede estar indicando que NGC 2682 tiene en realidad un *halo* más extenso de lo que en principio se esperaba. Algo realmente factible si se tiene en cuenta que la edad de este cúmulo, (unos 3 600 millones de años), es suficiente para haber disgregado ya buena parte de sus miembros iniciales. En este punto surge la idea de repetir el análisis inicial ampliando la definición del campo de cielo a estudiar hasta 5º o 6º.

Siguiendo con el estudio de resultados, Clusterix nos deja claro que el análisis que realiza se basa exclusivamente en los movimientos propios de las estrellas presentes en campo definido. No entra a 46

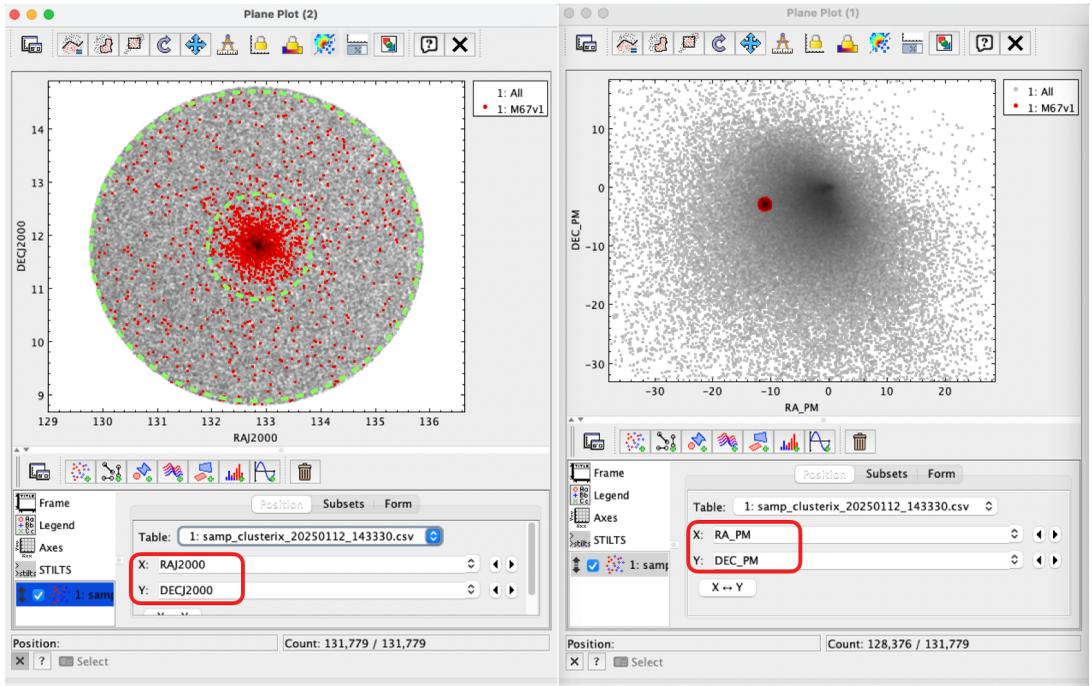


Figura 9. Izq. - posición en un mapa de coordenadas de las estrellas candidatas a formar parte del cúmulo (en rojo), según Clusterix, sobre el fondo del total de objetos extraídos de Gaia/EDR3 para esta región del cielo. Acotadas, con línea de puntos en verde la región 'C' (radio = 1°), y el campo total considerado inicialmente (3°). Dcha. - mapa en el espacio de configuración de movimientos propios. Las funciones de frecuencia analizadas por Clusterix han identificado la sobre-densidad con la que ha hecho su propuesta de membresía.

considerar otras propiedades como la *distancia* de estos objetos, por lo que el conjunto de miembros propuestos puede estar contaminado por estrellas que no formen parte del cúmulo. Es de esperar que los miembros del *cluster* cumplan con la propiedad de estar próximos entre sí, (razonablemente). Por tanto se hace necesario ver qué pasa con el paralaje (Fig. 10).

Al igual que hicimos en el artículo anterior, crearemos un nuevo subconjunto, 'M67v2', filtrando sobre M67v1, (aprovechamos para descartar también errores relativos superiores al 10 %):

$$\text{Expression} : M67v1 \ \&\& \ PLL < 1.14 + 0.5 \ \&\& \ PLL > 1.4 - 0.5 \ \&\& \ ePLL/PLL < 0.1 \quad (3.2)$$

Hemos aplicado el mismo criterio utilizado entonces para estrechar la selección al rango sobre el paralaje.

$$PLL \equiv \text{media} \pm \text{desviacion estandar} \quad (3.3)$$

Con esto el nuevo subconjunto 'M67v2' se reduce a 1 194 componentes que daremos por definitivo en el ejemplo desarrollado. En la Fig. 11 tenemos el cúmulo acotado en distancia: $869.5(pc) \pm 72(pc)$ ¹. En la parte derecha de la figura se representa el diagrama '*color (BP-RP) / magnitud*', (M67v1 en rojo y M67v2 en azul, sobre el fondo total del campo de cielo en gris), con un resultado similar al del primer artículo.

¹Recordemos que distancia (parsec) = 1000 / paralaje (mas)

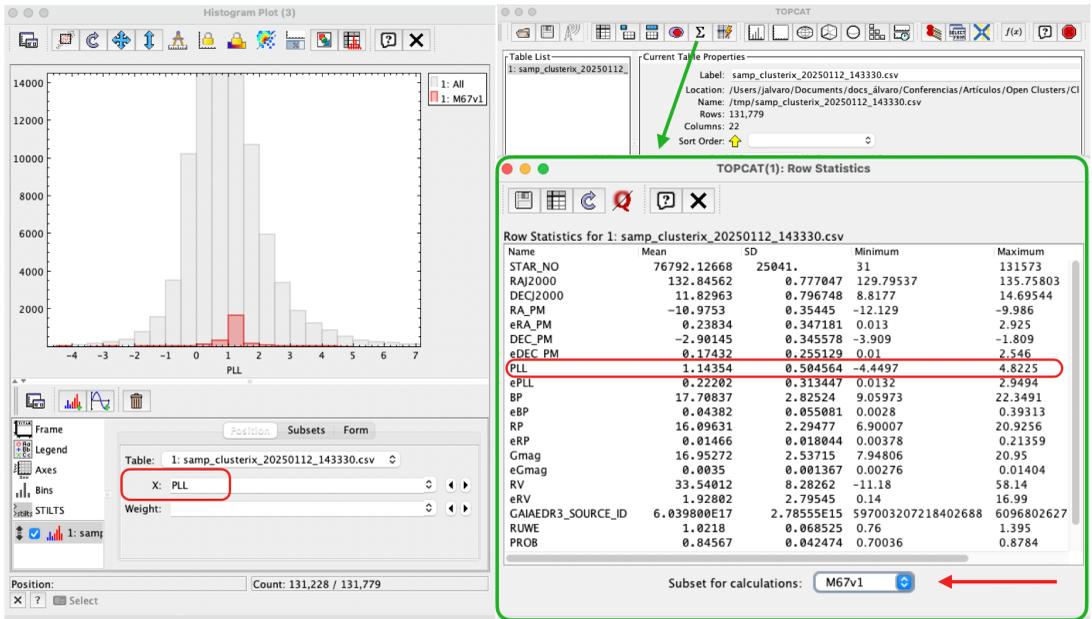


Figura 10. Izq. - Representación del histograma de paralajes. Se aprecia una dispersión en torno al valor medio que conviene estrechar. Dcha. - Datos estadísticos de los valores en el subconjunto 'M67v1', (miembros del cúmulo asignados por Clusterix). Se resaltan los correspondientes al paralaje.

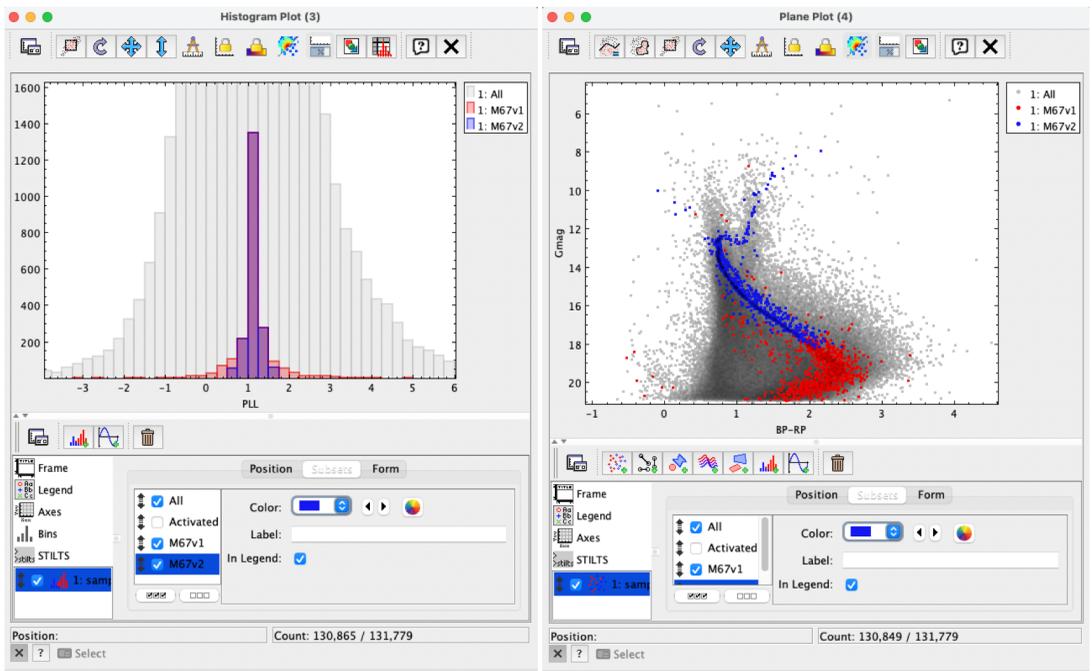


Figura 11. Izq. - Histograma de paralajes de los subconjuntos M67v1 y M67v2 sobre el fondo de estrellas de campo. Dcha. - Diagrama color-magnitud de ambos subconjuntos también sobre el fondo total.

4. Otras consideraciones

La Fig. 11 (dcha.) muestra las estrellas de M67v2 y M67v1 en el diagrama *color-magnitud*. Recordemos que M67v1 es el subconjunto propuesto por Clusterix y M67v2 es el resultado final que hemos adoptado una vez tradados con TOPCAT los paralajes. Buena parte de las estrellas de M67v1 (en rojo) están *tapadas* por los miembros de M67v2 (en azul) que están dibujados en una capa por encima.

Una primera conclusión es que el filtrado por paralaje produce un perfil bastante más nítido que el que presenta M67v1. Es especialmente notoria una amplia dispersión al nivel de las magnitudes más altas (parte inferior del gráfico).

Para entrar más en detalle en esta región vamos a generar un nuevo *sub-set* a partir de M67v1, en concreto seleccionaremos las estrellas con magnitud Gaia mayor que 19.

$$'M67v3' \equiv \text{Expression} : M67v1 \ \&\& \ Gmag > 19 \quad (4.1)$$

Comparamos algunos datos entre M67v2 y M67v3 en la Fig. 12. Como se aprecia en esta comparación

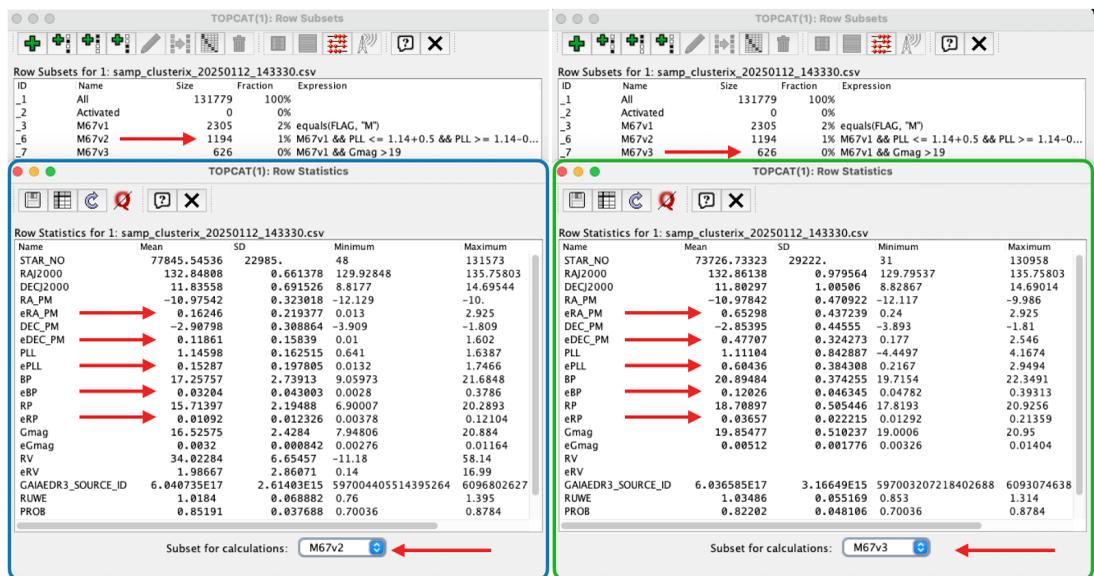


Figura 12. Izq. - Datos estadísticos sobre M67v2. Dcha. - Mismos datos relativos a M67v3. Señalados en la imagen los campos correspondientes a las incertidumbres empíricas sobre diversos parámetros. Son especialmente significativos los que afectan a los índices de color.

sobre ambas tablas, por encima de la magnitud 19 la precisión de Gaia, tanto en astrometría como en fotometría especialmente, se degrada de manera notoria. En la mayoría de ellos, (*movimientos propios* y *magnitudes BP* y *RP*), prácticamente esta incertidumbre se multiplica por 4. Con datos de Gaia/DR2 esto es aún más notorio.

La conclusión a extraer de aquí es que, a menos que ésta sea una región de interés expreso, conviene excluir de estos análisis los objetos más débiles del espectro de Gaia, (magnitud G de 19 en adelante).

Se propone al lector ejercitarse aplicando estos procesos con otros cúmulos como NGC 2516, NGC 1817, NGC 1750 y 1758, (todos ellos desarrollados en *L. Balaguer-Núñez et al. [2]*). También puede intentarse con Las Pleiades, (M45, *Melotte 22*).

En la próxima entrega haremos uso de otra herramienta del Observatorio Virtual: VOSA.

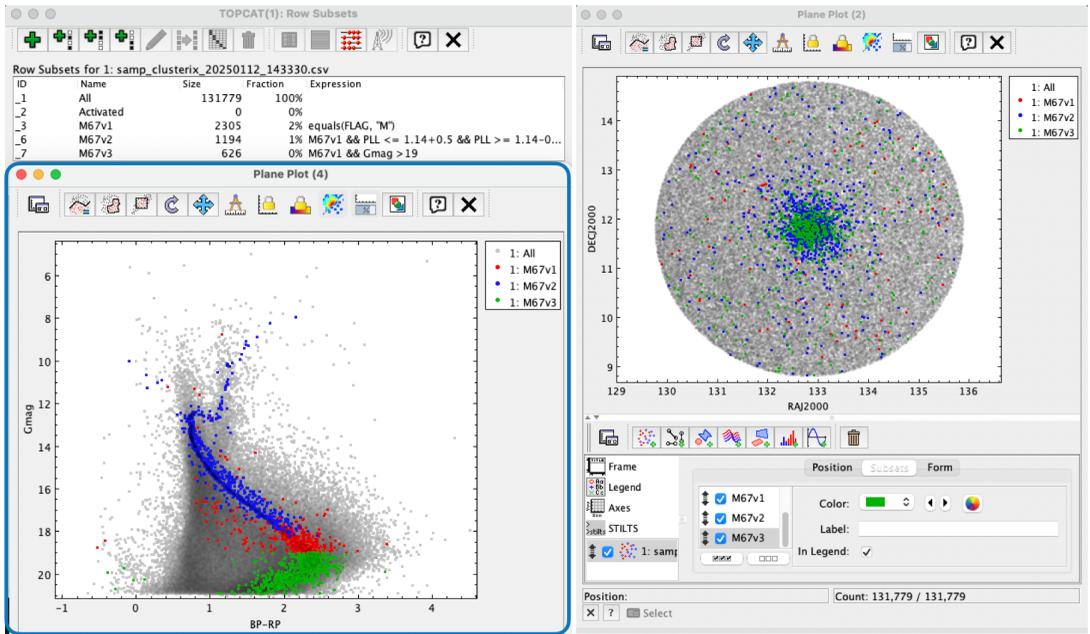


Figura 13. Izq. - Representación color-magnitud de todos los suconjuntos. Dcha. - Mapa de posiciones de los tres subconjuntos.

Agradecimientos

Clusterix 2.0 -

Clusterix 2.0 is an interactive web-based application to calculate the grouping probability of a list of objects using proper motions and the non parametric method in the approach described in Balaguer-Núñez et al (2020)

<http://clusterix.cab.inta-csic.es/clusterix/>

TOPCAT -

This research has made use of "TOPCAT VO"

Author: Mark Taylor, Astrophysics Group, Physics Department, University of Bristol

Email: m.b.taylor@bristol.ac.uk

TOPCAT WWW page: <http://www.starlink.ac.uk/topcat/>

<https://ui.adsabs.harvard.edu/abs/2005ASPC..347...29T/abstract>

Spanish Virtual Observatory-

This research has made use of the Spanish Virtual Observatory <https://svo.cab.inta-csic.es> project funded by MCIN/AEI/10.13039/501100011033/ through grant PID2020-112949GB-I00.

References

[1] Clusterix 2.0 - <http://clusterix.cab.inta-csic.es/clusterix/>

- [2] L. Balaguer-Núñez et al. - "Clusterix 2.0. A Virtual Observatory tool to estimate cluster membership probability" - <https://arxiv.org/pdf/1910.07356>
- [3] TOPCAT: Tool for OPERations on Catalogues And Tables - <https://www.star.bris.ac.uk/~mbt/topcat/>
- [4] Aladin Sky Atlas - <https://aladin.cds.unistra.fr>
- [5] VO SED Analyzer - <http://svo2.cab.inta-csic.es/theory/vosa/index.php>
- [6] D. Galadí-Enríquez et al. - "The overlapping open clusters ..." - Astronomy and Astrophysics, v.337, p.125-140 (1998) <https://ui.adsabs.harvard.edu/abs/1998A%26A...337..125G/abstract>
- [7] Open Clusters - CDS VizieR Catalogue - <http://vizier.cds.unistra.fr/viz-bin/VizieR-3?-source=B/ocl/clusters>
- [8] Misión Gaia (ESA) - <https://sci.esa.int/web/gaia/>
- [9] Spanish Virtual Observatory <https://svo.cab.inta-csic.es>. Escuelas y tutoriales: <https://svo.cab.inta-csic.es/docs/index.php?pagename=Meetings>

SECTION: PROGRAMACIÓN DE DISPOSITIVOS ASTRONÓMICOS

Diseño, construcción y programación de dispositivos astronómicos: Construcción física, electrónica y *firmware*

Sergio Alonso¹ and Javier Flores²

¹ Dpto. de Lenguajes y Sistemas Informáticos, Universidad de Granada ; Sociedad Astronómica Granadina ; España. E-mail: zerjtoi@ugr.es.

² Observatorio Astronómico de Calar Alto ; Sociedad Astronómica Granadina ; España. E-mail: jflores@caha.es.

Keywords: dispositivos astronómicos, *astronomical devices*, *hardware*, *software*, controlador, *driver*, INDI, INDIGO, panel de *flats*, *flats panel*

© Este artículo está protegido bajo una licencia [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/)

Resumen

En esta segunda entrega de la serie dedicada al diseño, construcción y programación de dispositivos astronómicos, se detallan los componentes mecánicos y electrónicos que se emplearán como referencia para la implementación de un panel de *darks/flats*. Asimismo, se presenta el diseño del circuito electrónico, el protocolo de comunicación y el desarrollo del *firmware* correspondiente que será la base para la creación de nuestro driver para INDIGO.

Abstract

In this second article of the series dedicated to the design, construction and programming of astronomical devices, the mechanical and electronic components that will be used as a reference for the implementation of a darks/flats panel are described in detail. It also presents the design of the electronic circuit, the communication protocol and the development of the corresponding firmware, which will be the basis for the creation of our future INDIGO driver.

Continuación de la serie

En el primer artículo de la serie [1] presentamos los conceptos generales necesarios para poder desarrollar un dispositivo astronómico desde cero. En este artículo nos centraremos en la construcción de la parte física, electrónica y la programación de su *firmware*. En posteriores artículos desarrollaremos un driver INDIGO que nos permita controlar el dispositivo desde cualquier programa compatible con este protocolo, y en el último artículo de la serie programaremos algunos clientes que hagan uso específico del mismo.

El dispositivo que queremos desarrollar consiste en una tapadera para un telescopio (que evitará el polvo mientras no se utiliza la óptica) a la que le hemos añadido un panel luminoso que nos permitirá efectuar tomas de *flats* de manera automática. Además, se le ha añadido un sencillo (y no muy preciso, todo hay que decirlo) sensor de temperatura. Este dispositivo pretende ser didáctico más que plenamente funcional por lo que hemos simplificado su diseño y programación al máximo. Se deja como tarea para el lector interesado el pensar e implementar distintas mejoras como pueden ser: atenuación de la luminosidad del panel *flat*, control de la velocidad de apertura y cierre de la tapa, incorporar sensores más precisos o de otra naturaleza, guardado de los parámetros de configuración en la memoria EEPROM del microcontrolador, etc.

Para el desarrollo del dispositivo, en las siguientes secciones detallaremos los siguientes aspectos:

- Diseño de las piezas físicas del dispositivo.
- Definición del protocolo de comunicación que usará para comunicarse con el ordenador al que se conecte.
- Diseño de la electrónica del dispositivo (basada en la plataforma de microcontroladores Arduino).
- Programación del *firmware* del dispositivo.
- Primeras pruebas de comunicación y control usando el puerto serie de Arduino.

1. Introducción: Creación de dispositivos astronómicos

Los dispositivos astronómicos llevan con nosotros desde que empezamos a mirar hacia las estrellas. Nuestros ojos vieron en el cielo nocturno patrones que servirían para trazar mapas estelares a través de las constelaciones y los calendarios serían usados para controlar los cambios de estación y celebrar rituales. Con ello inevitablemente llegaron los primeros instrumentos astronómicos: dólmenes, gnomon, el astrolabio, etc. A lo largo de la historia ha habido grandes descubrimientos que nos han llevado un poco más lejos en el conocimiento del Universo, desde el telescopio de Galileo y el prisma de Newton hasta la aparición de placas fotográficas y CCDs, que aunque ofrecen una manera distinta de observar el cielo, nos han permitido entender la naturaleza de todos esos puntos brillantes que tanto nos fascinan.

En la actualidad, cuando hablamos de instrumentación astronómica tenemos que distinguir dos grandes grupos. Por un lado nos encontramos con el telescopio y todo lo asociado a él: óptica del propio telescopio, sistemas de movimiento en Alpha (AR) y Delta (DEC), sistemas para controlar el foco, sistemas para controlar la rotación del instrumento, sistemas para controlar cúpula y casetas, etc. Por otro lado tenemos el conjunto de instrumentos encargados de recoger la luz. Dependiendo de como queramos analizar esa luz necesitaremos unos dispositivos u otros. Por lo tanto distinguiremos entre telescopio y todo lo asociado a él y los instrumentos que recolectan la luz.

Desde el punto de vista de la instrumentación astronómica profesional al diseñar un instrumento, en la mayoría de los casos, se diseña totalmente a medida. Esto quiere decir que cada una de las piezas de ese instrumento (cámara, espectrógrafo, unidades de calibración, sistemas ópticos, etc) son cuidadosamente diseñados o, en el caso de detectores, elegidos para ese instrumento.

El diseño de un instrumento normalmente va ligado a un proyecto científico, por lo que tenemos que conocer los objetos que serán fruto de su estudio. También necesitamos conocer cuales son el resto de instrumentos disponibles en el mundo para crear algo novedoso que incremente la importancia y utilidad científica del instrumento diseñado. Por último, necesitamos conocer las limitaciones tecnológicas de la época en la que se construye el instrumento. Por lo tanto, en la construcción de instrumentación, aunque cada proyecto lleva uno o varios Investigadores Principales (PI) asociados, el nivel de participación de personas en distintas áreas de conocimiento es muy grande.

En el caso de la astronomía amateur normalmente no vamos a diseñar un instrumento completo, sino que optaremos por soluciones prácticas adaptadas a las necesidades de nuestro equipo particular; la razón es que, tanto a nivel profesional como aficionado, la creación de dispositivos astronómicos responde a la escasa comercialización o fabricación en serie de muchas unidades. Mientras que en el ámbito profesional el coste se prevé previamente y se integra en el presupuesto, en el amateur el factor económico resulta decisivo, pues aunque a veces existan en el mercado los accesorios que necesitamos, su producción limitada eleva notablemente los precios en comparación con el coste de los materiales. Por eso, aprovechando las nuevas tecnologías —impresión 3D (FDM y resina), corte láser, máquinas CNC, electrónica de bajo coste, etc.- podemos fabricar nosotros mismos las piezas e instrumentos que realmente necesitamos.

Para la creación de dispositivos existen varias alternativas: Por un lado podemos elegir un proyecto que

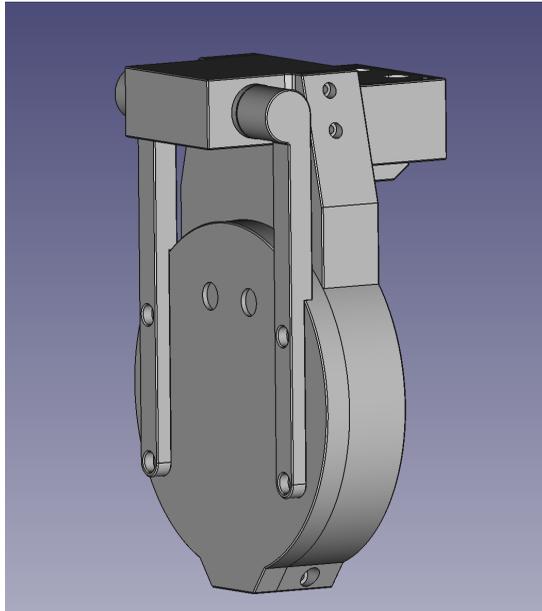


Figura 1. Modelo 3D del panel de flats ASCOM-Compatible Telescope Cover.

alguien ha desarrollado previamente e intentar reproducirlo, ya sea con mejoras o adaptado a nuestro caso particular o por otro lado podemos diseñar el dispositivo desde cero y completamente, desde el prototipado y montaje mecánico de las piezas hasta la programación de la electrónica.

2. Diseño de las piezas físicas del dispositivo

Para crear nuestro dispositivo debemos construir una serie de piezas que compongan el armazón donde se incrustarán los componentes mecánicos y electrónicos del mismo (panel LED, motor servo, etc). Para esta tarea, y por no reinventar la rueda, hemos optado por reutilizar un diseño libre existente en la red llamado ASCOM-Compatible Telescope Cover[2], creado por Julien Lecomte. Este diseño puede modificarse utilizando programas libres de diseño 3D paramétrico como es, por ejemplo, FreeCAD [3]. Existen numerosos recursos en línea sobre el uso y manejo de este software[4, 5, 6]. Podemos ver el modelo 3D desde FreeCAD en la Fig. 1. Es importante notar que de dicho proyecto solo vamos a utilizar el diseño de las piezas, ya que la electrónica, el *firmware*, *driver*, etc. será íntegramente desarrollado por nosotros.

Las piezas de dicho diseño pueden ser adaptadas ya que el autor de dicho diseño incorporó algunos parámetros (como la apertura del telescopio) para ajustar el dispositivo a las necesidades particulares de cada usuario, tal y como podemos ver en la pestaña *parameter* (Fig. 2) de FreeCAD.

Una vez ajustado el diseño a nuestras necesidades específicas se pueden imprimir las piezas con una impresora 3D. Para ello hay que exportar cada una de las piezas desde FreeCAD a un formato legible por el programa que usemos para poder hacer el loncheado (*slicing*), como podemos ver en la figura 3 y que permitirá a nuestra impresora producirlas. Lo más común para esta tarea es usar el formato STL, aunque existen otros. El material que hemos usado para la impresión es PLA, pero pueden usarse otros materiales como PETG, ABS, etc.

Una vez impresas, usaremos algunos tornillos, arandelas, tuercas y roscas insertables para ensamblar todas las piezas como podemos ver en la figura 4. En particular, para este diseño, se han utilizado:

- 4 tornillos M3*5mm + 4 tuercas M3

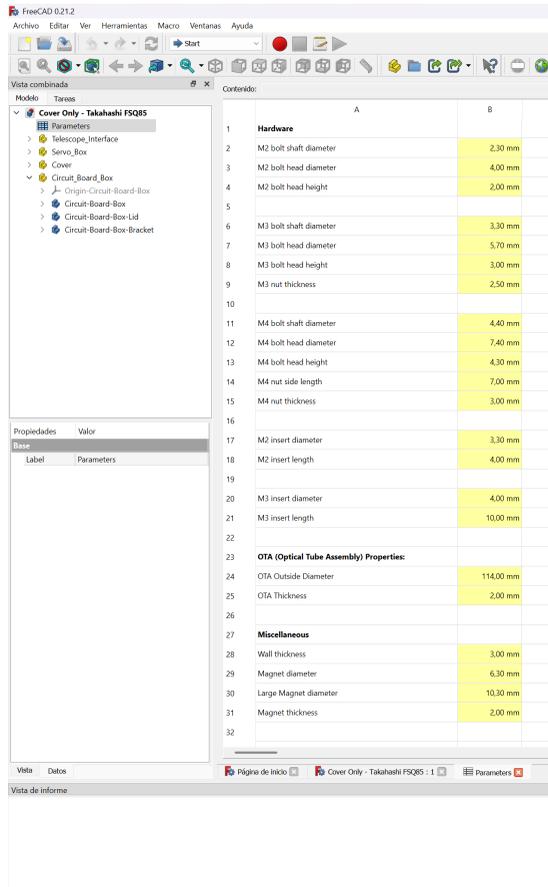


Figura 2. Parámetros modificables en FreeCAD del ASCOM-Compatible Telescope Cover.

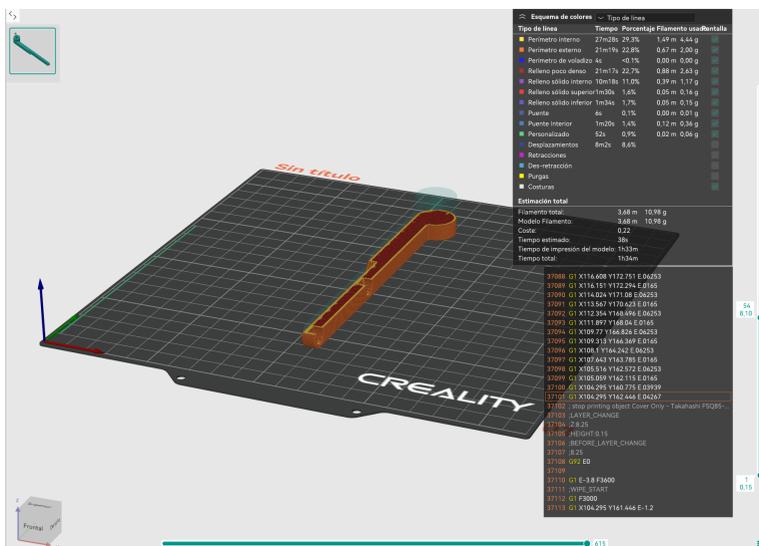


Figura 3. Programa de slicing (OrcaSlicer) preparando un brazo del panel de flats.



Figura 4. Piezas impresas en PLA y montadas.

- 4 tornillos M3*8mm + 4 roscas insertables M3
- 4 tornillos M3*10mm + 4 roscas insertables M3
- 4 tornillos M3*10mm + 4 tuercas M3
- 3 tornillos M3*10mm + 3 tuercas M3
- 4 tornillos M3*25mm + 4 arandelas M3
- 1 tornillo M4*30mm + 1 tuerca M4

Hemos puesto a disposición del lector dos modelos con los archivos en formato STL que se pueden descargar desde Github: <https://github.com/zerjillo/TelescopeFlatsCover/tree/main/STL>. Estos modelos son para un refractor Skywatcher Esprit 120ED y para un refractor Askar103APO.

3. Protocolo de comunicación

El protocolo de comunicación consiste en la especificación de los comandos (mensajes) que intercambiarán el ordenador y la electrónica de nuestro dispositivo a través del puerto serie (USB). Cuando se diseña un dispositivo es muy importante definir de manera clara este protocolo para poder luego implementar drivers u otros programas que quieran controlar directamente el dispositivo. Estas instrucciones deben permitir obtener toda la información sobre el dispositivo así como activar cada uno de los actuadores del mismo. Sin embargo, estos comandos deben ser lo más simples posible, dejando que de la lógica de alto nivel se encarguen los drivers o incluso los programas clientes que los usen.

Por ejemplo, para nuestro dispositivo debemos contar con algunos mensajes que permitan abrir o cerrar

Comando	Descripción	Respuesta	Ejemplo
VERSION	Pregunta por la versión del <i>firmware</i> del dispositivo	Mens. VERSION	
STATUS	Pide la información de estado del dispositivo (abierto, cerrado, encendido, apagado...)	Mens. STATUS	
OPEN	Pide que se abra la tapa del dispositivo	Mens. STATUS	
CLOSE	Pide que se cierre la tapa del dispositivo	Mens. STATUS	
LON	Pide que se encienda el panel de <i>flats</i>	Mens. STATUS	
LOFF	Pide que se apague el panel de <i>flats</i>	Mens. STATUS	
START	Pide que se comiencen a mandar mensajes de STATUS cada cierto intervalo	Mens. STATUS	
STOP	Pide que se pare de mandar mensajes de STATUS cada cierto intervalo	Mens. STATUS	
INTERVAL INT	Establece el intervalo de los mensajes de STATUS. El parámetro es un entero entre 1 y 3600 segundos	Mens. STATUS	INTERVAL 30 Fijará el intervalo de mensajes a 30 segundos.

Cuadro 1. Comandos hacia el dispositivo del protocolo de comunicación.

Respuesta	Descripción	Ejemplo
VERSION	Devuelve la versión del <i>firmware</i> del dispositivo	VERSION 1.00b
STATUS	Devuelve la información de estado del dispositivo: <ul style="list-style-type: none"> ■ cerrado o abierto (0/1), ■ apagado o encendido el panel de <i>flats</i> (0/1), ■ temperatura (grados Celsius), ■ mandando mensajes de STATUS periódicamente o no (1/0), ■ intervalo de los mensajes (segundos) 	STATUS 0 1 23.4 1 15 <ul style="list-style-type: none"> ■ tapa está cerrada, ■ panel de <i>flats</i> encendido, ■ el sensor detecta 23.4 °C, ■ se están mandando mensajes de STATUS periódicamente, ■ cada 15 segundos

Cuadro 2. Mensajes de respuesta del protocolo de comunicación.

la tapa, encender o apagar el panel de *flats*, pero no deberíamos incorporar ningún mensaje que, por ejemplo, automatizase el cierre de la tapa, el encendido del panel durante un tiempo y luego volviese a apagarlo y abrirlo. De un control complejo como el descrito debería hacerse cargo el software del ordenador de control.

En la tabla 1 se muestran los mensajes (comandos) que atenderá nuestro dispositivo (es decir, que puede mandarle el ordenador de control). Cada uno de esos comandos se responderá con otro mensaje de vuelta tal y como se describen en la tabla 2.

Como se deduce de dichas tablas, todos los comandos, exceptuando el comando VERSION se responden con un mensaje STATUS. Ante un mensaje erróneo o inexistente el dispositivo lo ignorará (no responderá nada). Además, cuando el dispositivo se enciende, enviará una única vez la siguiente cadena de caracteres por el puerto serie hacia el ordenador: Telescope Flat Cover connected and initialized.

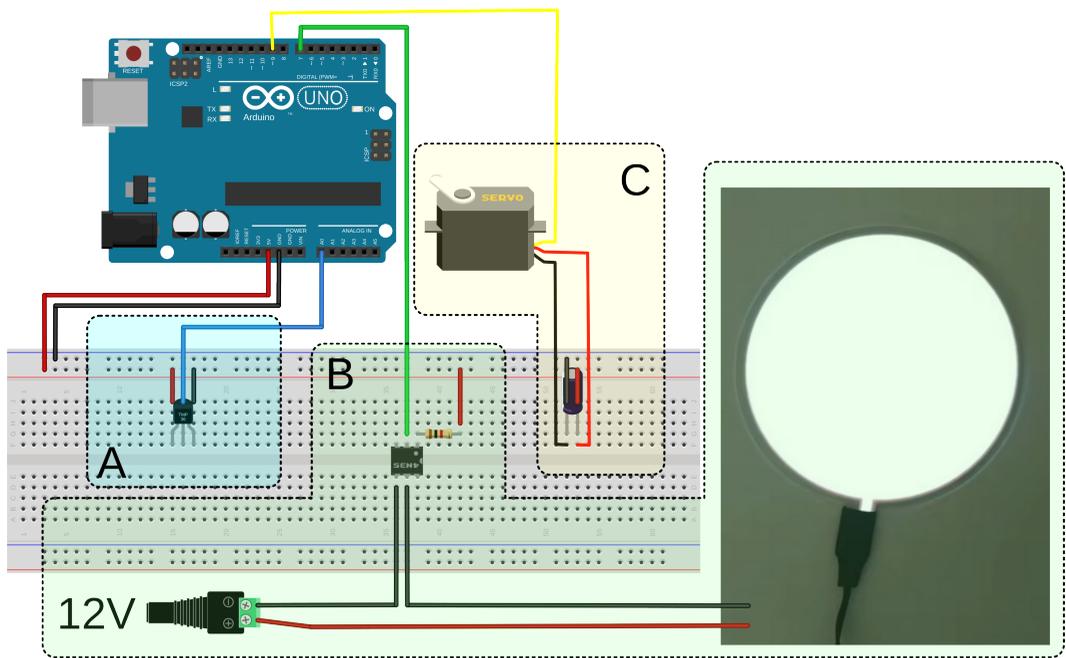


Figura 5. Esquema de conexión de los componentes electrónicos.

4. Diseño de la electrónica del dispositivo

Para la electrónica del dispositivo proponemos el uso de la plataforma Arduino [7]. Esta plataforma surgió como un intento de hacer accesible la programación de microcontroladores. Pese al carácter didáctico con el que nació se ha convertido en una plataforma ampliamente utilizada para proyectos de todo tipo. Existen numerosos manuales y tutoriales al respecto en la red [8].

Arduino en realidad se refiere hoy en día a todo un ecosistema de distintas placas electrónicas con un microcontrolador acoplado que permiten interconectar diferentes sensores y actuadores, junto con otros componentes como un puerto USB para programarlo y comunicarlo con el exterior. Para un proyecto como el nuestro se puede usar cualquier placa de la familia de Arduino, pero para los esquemas y prototipos hemos utilizado una placa Arduino UNO. Para el proyecto final podría usarse otra placa más pequeña a la que soldarle los componentes (en vez de montarlos en una placa de prototipado como en las imágenes que acompañan este trabajo). Para programar el *firmware* existe el entorno de programación (IDE) de Arduino, que puede descargarse gratuitamente de la página oficial [7]. Los componentes principales que usaremos en este proyecto (todos bastante baratos y fáciles de conseguir en línea o en tiendas de electrónica) son:

- Placa Arduino UNO
- Sensor de temperatura TMP36
- Optoacoplador 4N35
- Resistencia de 1kΩ.
- Un servomotor
- Un condensador (100μf)
- Una fuente de alimentación de 12V (para el panel de flats)
- Panel de flats redondo.

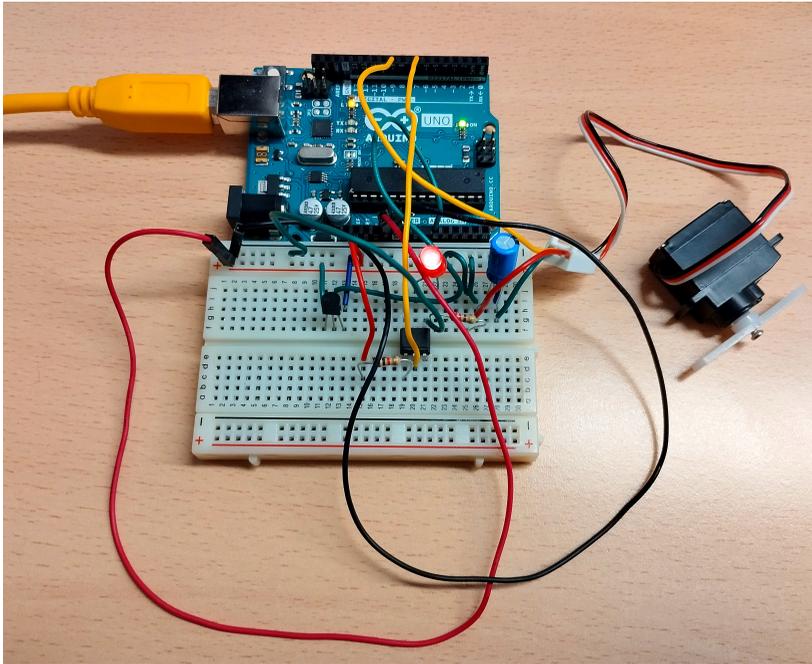


Figura 6. Montaje de Arduino UNO.

Dichos componentes se conectan entre sí de acuerdo con el esquema de la figura 5. El circuito tiene 3 bloques principales:

- **Bloque A:** Contiene el sensor de temperatura TMP36. Dicho sensor tiene tres patillas: alimentación a 5V, tierra (GND) y la intermedia que conectaremos a la entrada analógica A0 del Arduino.
- **Bloque B:** Contiene los componentes para controlar el panel de flats. Como dicho panel funciona a un voltaje distinto del Arduino (12V frente a los 5V de Arduino) utilizamos un optoacoplador 4N35 que permite aislar el circuito del panel de flats. Simplificando mucho, un optoacoplador funciona como un interruptor que puede ser activado por una señal eléctrica. En nuestro caso cierra o abre el circuito de 12V del panel según sea la señal que le llega desde el pin 7 del Arduino.
- **Bloque C:** Contiene los componentes para controlar el servo motor. Un servomotor es un tipo de motor que puede controlar el ángulo de giro de su vástago según una señal que le proporcionemos desde el microcontrolador (en nuestro caso a través del pin 9 del Arduino). Dado que los motores pueden introducir algunas señales eléctricas en la circuitería de Arduino se recomienda usar un condensador de 100µf entre los dos polos de alimentación del servo para atenuar las posibles interferencias.

En la Fig. 6 se puede ver una foto del prototipo mientras se estaba desarrollando, donde el panel de flats se estaba aún probando como un simple led rojo.

5. Programación del *firmware* del dispositivo

Arduino tiene un lenguaje de programación propio [9] con una sintaxis muy parecida a C y C++. El lenguaje incorpora además los mecanismos necesarios para activar los distintos pines de entrada/salida de la placa (I/O) tanto como para la lectura de señales analógicas (en los pines A0-A6), control del puerto serie del microcontrolador (junto con el conversor a puerto USB, que es el que realmente conectaremos al ordenador), así como una gran cantidad de bibliotecas que simplifican el control de muchos motores

(de corriente continua, paso a paso, servos...), actuadores y sensores existentes en el mercado.

A continuación describiremos las secciones de código principales del *firmware* con el que programaremos nuestra placa Arduino para poder controlar los distintos componentes del dispositivo. Dicho código completo puede descargarse del siguiente repositorio de Github: <https://github.com/zerjillo/TelescopeFlatsCover>.

Para el desarrollo del *firmware* hacemos uso de dos bibliotecas de Arduino. La primera se llama `CommandParser` [10] y tiene como finalidad facilitar el análisis de comandos que se reciban en el microcontrolador (por ejemplo a través del puerto serie). Esta biblioteca no está instalada en el IDE de Arduino por defecto, por lo que es posible que para usarla haga falta instalarla desde **Herramientas -> Gestionar bibliotecas...** del entorno de programación.

Esta biblioteca permite definir una serie de comandos de texto, con parámetros opcionales, de tal manera que cuando se recibe uno de dichos mensajes se llamará a una función *callback* que previamente habremos definido. De esta manera el análisis (*parsing*) de los comandos se simplifica enormemente. En el listado 1 se muestran las secciones de código fundamentales que tienen que ver con el análisis de los comandos recibidos por Arduino. Como se puede apreciar definimos una serie de funciones *callback* (las hemos llamado todas con el prefijo `cmd_`). Cada una de esas funciones incluye el código necesario para responder ante un comando particular recibido. En la función `setup()` de Arduino, que se ejecuta una única vez cada vez que la placa se enciende y se utiliza para inicializar la placa, activamos e inicializamos el puerto serie y registramos cada uno de los comandos que la placa debe reconocer y la función *callback* a la que tendría que llamar. Así por ejemplo, la instrucción:

```
parser.registerCommand("INTERVAL", "i", &cmd_updateInterval);
```

indica que si recibimos el mensaje `INTERVAL INT`, donde `INT` es un parámetro entero (`i`) debe llamarse a la función *callback* `cmd_updateInterval()` definida anteriormente. Por último, para gestionar la recepción de comandos, en la función `loop()` (que se está ejecutando de manera iterativa durante todo el tiempo que la placa está encendida) comprobamos si existen mensajes por leer en el puerto serie y, en caso afirmativo, le mandamos a la biblioteca `CommandParser` que analice el mensaje y llame a la función *callback* apropiada:

```
parser.processCommand(line, response);
```

```
#include <CommandParser.h> // Incluimos la biblioteca CommandParser, previamente instalada en el Arduino IDE
...
typedef CommandParser<> MyCommandParser; // Objeto CommandParser
MyCommandParser parser;
...
// Funciones callback que atienden cada uno de los comandos que se pueden recibir
void cmd_version(MyCommandParser::Argument *args, char *response) { ... }
void cmd_status(MyCommandParser::Argument *args, char *response) { ... }
void cmd_open(MyCommandParser::Argument *args, char *response) { ... }
void cmd_close(MyCommandParser::Argument *args, char *response) { ... }
void cmd_lightOn(MyCommandParser::Argument *args, char *response) { ... }
void cmd_lightOff(MyCommandParser::Argument *args, char *response) { ... }
void cmd_startUpdating(MyCommandParser::Argument *args, char *response) { ... }
void cmd_stopUpdating(MyCommandParser::Argument *args, char *response) { ... }
void cmd_updateInterval(MyCommandParser::Argument *args, char *response) { ... }
...
// La funcion setup() se ejecuta una unica vez cuando la placa Arduino se enciende
void setup() {
  Serial.begin(9600); // Inicializamos el puerto Serie
  while (!Serial);
  ...
  // Registramos todos los comandos disponibles
  parser.registerCommand("VERSION", "", &cmd_version);
  parser.registerCommand("STATUS", "", &cmd_status);
  parser.registerCommand("OPEN", "", &cmd_open);
  parser.registerCommand("CLOSE", "", &cmd_close);
  parser.registerCommand("LON", "", &cmd_lightOn);
}
```

```

parser.registerCommand("LOFF", "", &cmd_lightOff);
parser.registerCommand("START", "", &cmd_startUpdating);
parser.registerCommand("STOP", "", &cmd_stopUpdating);
parser.registerCommand("INTERVAL", "i", &cmd_updateInterval); // Con un parametro entero entre 1 y 3600
...
}

// La funcion loop() se ejecuta continuamente (en un bucle) mientras que la placa Arduino este encendida
void loop() {
  if (Serial.available()) { // Si hay mensajes por leer en el puerto serie
    char line[128];
    size_t lineLength = Serial.readBytesUntil('\n', line, 127); // Read serial port
    line[lineLength] = '\0';

    char response[MyCommandParser::MAX_RESPONSE_SIZE];
    parser.processCommand(line, response); // Analiza el mensaje y llama a la funcion callback correspondiente
  }
  ...
}

```

Listado 1. Código del firmware relacionado con el análisis de los comandos.

Para el control del servo motor hacemos uso de la biblioteca Servo (listado 2). Esta biblioteca permite de manera sencilla especificar el ángulo en el que debe posicionarse el vástago del motor llamando a la función:

```
servo.write(angulo);
```

En nuestro caso esa función la llamamos desde la función setActuators() que es la que se encarga de comprobar la posición en la que debe estar la tapa (variable coverOpen que habrá sido establecida por las funciones *callback* al recibirse los mensajes OPEN o CLOSE).

```

...
#include <Servo.h>
...
const int SERVO_PIN = 9;
const int CLOSED_ANGLE = 0;
const int OPENED_ANGLE = 90;
...
bool coverOpen = false; // Esta abierta la tapa?
...
Servo servo; // El objeto para controlar el motor
...
void setActuators() {
  if (coverOpen) {
    servo.write(OPENED_ANGLE);
  } else {
    servo.write(CLOSED_ANGLE);
  }
  ...
}
...
void setup() {
  ...
  servo.attach(SERVO_PIN); // Inicializamos la biblioteca servo con el pin al que lo tenemos conectado
  ...
}
...

```

Listado 2. Código del firmware relacionado con el control del motor servo.

El control del encendido o apagado del panel de *flats* es más sencillo, pues solo tenemos que activar o no el pin donde hemos conectado el optoacoplador (listado 3). Ese pin lo activamos o desactivamos también desde la función setActuators().

```

...
const int OPTO_PIN = 7;
...

```

```

bool lightOn = false; // Esta el panel encendido?
...
void setActuators() {
...
  if (lightOn) {
    digitalWrite(OPTO_PIN, LOW);
  } else {
    digitalWrite(OPTO_PIN, HIGH);
  }
...
}
...
void setup() {
...
  pinMode(OPTO_PIN, OUTPUT); // Inicializamos el optoacoplador par aencender y apagar el panel
...
}
...

```

Listado 3. Código del firmware relacionado con el encendido y apagado del panel de flats.

La obtención de la temperatura desde el sensor TMP36 es también simple: solo hay que hacer una lectura del pin analógico donde se ha conectado dicho sensor y hacer algunas operaciones aritméticas para transformar el nivel de la señal eléctrica medida a la temperatura en °C (listado 4).

```

...
const int TEMP_SENSOR = A0;
...
float getTemperature() {
  int read = analogRead(TEMP_SENSOR);
  float temperature = (read * (500.0 / 1023.0)) - 50.0; // En grados Celsius
  return temperature;
}
...

```

Listado 4. Código del firmware relacionado con el sensor de temperatura.

6. Primeras pruebas de comunicación y control (puerto serie)

Para comprobar el correcto funcionamiento de nuestro dispositivo y de su *firmware* podemos utilizar el monitor serie del IDE de Arduino (Herramientas ->Monitor Serie) que nos abrirá un panel inferior desde el que podremos mandar comandos al microcontrolador así como ver los mensajes de respuesta del mismo. Podemos por tanto ir haciendo pruebas de cada uno de los comandos definidos y comprobando que los actuadores funcionan correctamente (el motor se mueve y el panel de *flats* se enciende y apaga). En la Fig. 7 se muestra el IDE de Arduino con el monitor serie activado y con algunos comandos mandados a nuestro dispositivo y algunas respuestas recibidas. Queda como ejercicio para el lector descubrir que comandos se le enviaron al dispositivo para obtener los mensajes de respuesta que se observan en la figura.

7. Conclusiones

En este trabajo se ha presentado la parte mecánica, electrónica (basada en la plataforma Arduino) y el *firmware* de un dispositivo astronómico consistente en una tapa para telescopio que incorpora un panel de *flats* y un sensor de temperatura. Además, se ha especificado el protocolo de comunicación que permite mandar comandos desde el ordenador donde se conecte el dispositivo así como las respuestas que el dispositivo realiza ante los comandos recibidos.

En los próximos artículos de la serie construiremos un driver INDIGO que nos permita controlar este instrumento desde cualquier programa de control de dispositivos astronómicos compatible con INDI o

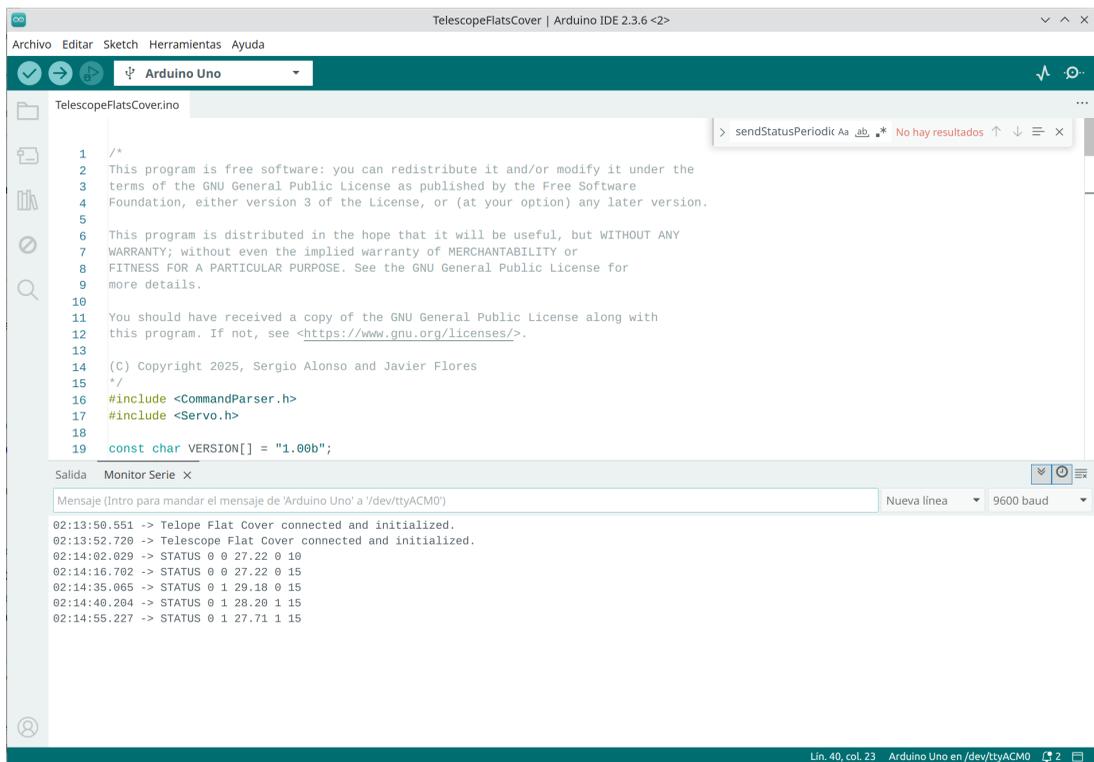


Figura 7. IDE de Arduino con el monitor serie activo para comprobar la comunicación y funcionamiento del dispositivo.

INDIGO así como directamente programar un cliente que haga uso del mismo dispositivo de manera programática.

Referencias

- [1] S. Alonso y J. Flores, *Diseño, construcción y programación de dispositivos astronómicos: una introducción*, JCAAC, **2**, 75–86 (2025).
- [2] Ascom-telescope-cover-v2. Disponible en <https://github.com/jlecomte/ascom-telescope-cover-v2>
- [3] FreeCAD. Tu modelador paramétrico 3D. Accesible y descargable en: <https://www.freecad.org/>
- [4] Impresión 3D. Dispositivos para observatorios en remoto - 1ª parte. Disponible en: https://www.youtube.com/watch?v=IIANo79QKM&ab_channel=FAAE
- [5] Impresión 3D. Dispositivos para observatorios en remoto - 2ª parte. Disponible en: https://www.youtube.com/watch?v=54UVxNeS8cQ&ab_channel=FAAE
- [6] Tutorial FreeCAD. J. González Gómez. Disponible en: https://www.youtube.com/watch?v=2_DbFzFV9D4.
- [7] Plataforma Arduino. Disponible en: <https://www.arduino.cc/>
- [8] Tutoriales de Arduino y otros microcontroladores por Luis Llamas. Disponibles en: <https://www.luisllamas.es/tutoriales-arduino/>
- [9] Arduino Language Reference. Disponible en: <https://docs.arduino.cc/language-reference/>
- [10] Biblioteca `CommandParser` para Arduino. Disponible en: <https://github.com/Uberi/Arduino-CommandParser>

SECTION: ASTRONOMICAL COMPUTING

Cálculo de efemérides

Tomás Alonso Albi¹¹Astrofísico en el Observatorio Astronómico Nacional, Spain. E-mail: talonsoalbi@gmail.com.**Keywords:** programación, programming, efemérides, ephemerides, cálculo astronómico, astronomical computing© Este artículo está protegido bajo una licencia [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/)Este artículo adjunta un *software* accesible en <https://github.com/JCAAC-FAAE>

Resumen

En este número veremos cómo reducir las coordenadas de un objeto para obtener la posición del mismo en el cielo para un observador situado sobre la superficie terrestre. Veremos el caso más sencillo en el que suponemos que ya tenemos la posición eclíptica geocéntrica del objeto referida al equinoccio medio de la fecha, de manera que no es necesario aplicar la corrección de precesión, vista en entregas anteriores. También se introducirá cierta cantidad de código adicional para tener bien organizado los resultados de los cálculos, incluyendo los resultados de operaciones potencialmente iterativas, como obtener los instantes de salida y puesta del objeto. En el caso de cuerpos estáticos, como las estrellas, esta iteración no es necesaria.

Abstract

In this article we see how to reduce the coordinates of an object to obtain its position in the sky for an observer located on the Earth's surface. We will examine the simplest case, in which we assume that we already have the geocentric ecliptic position of the object referred to the mean equinox of the date, so it is not necessary to apply the correction for precession, as discussed in previous articles in this Section. We also introduce some additional code to keep the calculation results well organized, including the results of potentially iterative operations, such as obtaining the rise and set times of the object. In the case of static bodies, such as stars, this iteration is not necessary.

1. Introducción

En esta entrega sólo se mostrará el cálculo práctico de la posición del Sol, con un algoritmo de baja precisión con un error máximo en torno a 10 segundos de arco. También veremos el cálculo de la posición aparente de una estrella estática a partir de las coordenadas J2000 de los catálogos. En próximas entregas veremos cómo aplicar el código presentado aquí para obtener posiciones muy precisas del Sol y la Luna.

Como siempre, el código presentado en esta ocasión puede también encontrarse en el repositorio de GitHub [1] correspondiente a esta sección, cuyo uso (y posible adaptación a otros lenguajes) ayudará a evitar los frecuentes errores de codificación del pasado.

2. Organización de los resultados del cálculo de efemérides

En el código presentado en esta sección se pretende dar una sugerencia de cómo organizar la información de salida del cálculo de efemérides. Se trata de una clase Java que debe utilizarse para instanciar o crear objetos que contendrán los resultados de las efemérides de los distintos cuerpos. Cada objeto contendrá valores particulares de campos como el acimut, la elevación, la ascensión recta, declinación, o distancia de un objeto particular para unas condiciones concretas del cálculo, que incluyen, entre otras opciones,

el instante de cálculo y la posición del observador. En otros lenguajes de programación populares como Python también es posible crear este tipo de objetos.

El código que se presenta contiene además algunas funciones útiles. El método *toString* está destinado a mostrar los resultados de los valores contenidos en el objeto, para reportar los resultados de las efemérides calculadas. El método *setIlluminationPhase* permite rellenar el valor del campo *illuminationPhase* (fase de iluminación) siempre que se proporcione como entrada la posición del Sol. Este método debe utilizarse tras obtener las efemérides, y sólo para objetos distintos del Sol. El método *getDateAsString* es un método auxiliar del método *toString* para formatear fechas. En él se utilizan los métodos estáticos *getHMS* y *fmt02* de la clase *Util*, los cuales pueden consultarse en el repositorio. En el primero se proporciona la hora, minuto, y segundo de una fracción de día expresado en radianes, y en el segundo se formatea con dos dígitos un valor numérico, añadiendo un cero delante si el valor es menor a 10.

Es técnicamente posible prescindir de este tipo de código en la organización de la información de salida, pero poco recomendable, pues a medida que avancemos esto supondría, muy probablemente, tener un código menos limpio, y con más funciones repetidas en distintos ficheros.

```
1 package journal;
2
3 /**
4  * A class to hold the results of the ephemerides calculations.
5  */
6 public class EphemData {
7
8     /** Values for azimuth, elevation, rise, set, and transit for the Sun. Angles in
9         radians, rise ...
10    * as Julian days in UT. Distance in AU */
11    public double azimuth, elevation, rise, set, transit, transitElevation, distance,
12        rightAscension,
13        declination, illuminationPhase, eclipticLongitude, eclipticLatitude,
14        angularRadius;
15
16    /**
17     * Main constructor
18     * @param azi Azimuth
19     * @param ele Elevation
20     * @param rise2 Rise
21     * @param set2 Set
22     * @param transit2 Transit
23     * @param transit_alt Transit elevation
24     * @param ra Right ascension
25     * @param dec Declination
26     * @param dist Distance
27     * @param eclLon Ecliptic longitude
28     * @param eclLat Ecliptic latitude
29     * @param angR Angular radius
30     */
31    public EphemData(double azi, double ele, double rise2, double set2,
32        double transit2, double transit_alt, double ra, double dec,
33        double dist, double eclLon, double eclLat, double angR) {
34        azimuth = azi;
35        elevation = ele;
36        rise = rise2;
37        set = set2;
38        transit = transit2;
39        transitElevation = transit_alt;
40        rightAscension = ra;
41        declination = dec;
42        distance = dist;
43        illuminationPhase = 100;
44        eclipticLongitude = eclLon;
45        eclipticLatitude = eclLat;
```

```

43     angularRadius = angR;
44 }
45
46 @Override
47 public String toString() {
48     String degSymbol = "\u00b0";
49     String lsep = Util.getLineSeparator();
50     StringBuilder sb = new StringBuilder();
51     sb.append(" Az:   "+(float) (azimuth * Constant.RAD_TO_DEG)+degSymbol + lsep);
52     sb.append(" El:   "+(float) (elevation * Constant.RAD_TO_DEG)+degSymbol +
53         lsep);
54     sb.append(" Dist: "+(float) (distance)+" au" + lsep);
55     sb.append(" RA:   "+(float) (rightAscension * Constant.RAD_TO_DEG)+degSymbol +
56         lsep);
57     sb.append(" DEC:  "+(float) (declination * Constant.RAD_TO_DEG)+degSymbol +
58         lsep);
59     sb.append(" Ill:  "+(float) (illuminationPhase)+"%" + lsep);
60     sb.append(" ang.R: "+(float) (angularRadius * Constant.RAD_TO_DEG)+degSymbol +
61         lsep);
62     sb.append(" Rise: "+EphemData.getDateAsString(rise) + lsep);
63     sb.append(" Set:  "+EphemData.getDateAsString(set) + lsep);
64     sb.append(" Transit: "+EphemData.getDateAsString(transit)+" (elev. "+(float)
65         (transitElevation * Constant.RAD_TO_DEG)+degSymbol+")" + lsep);
66     return sb.toString();
67 }
68
69 /**
70  * Sets the illumination phase field for the body
71  * @param sun The Ephem object for the Sun
72  */
73 public void setIlluminationPhase(EphemData sun) {
74     double dlon = rightAscension - sun.rightAscension;
75     double cosElong = (Math.sin(sun.declination) * Math.sin(declination) +
76         Math.cos(sun.declination) * Math.cos(declination) * Math.cos(dlon));
77
78     double rsun = sun.distance;
79     double rbody = distance;
80     // Use elongation cosine as trick to solve the rectangle and get rp (distance
81     // body - sun)
82     double rp = Math.sqrt(-(cosElong * 2.0 * rsun * rbody - rsun * rsun - rbody *
83         rbody));
84
85     double dph = ((rp * rp + rbody * rbody - rsun * rsun) / (2.0 * rp * rbody));
86     illuminationPhase = 100 * (1.0 + dph) * 0.5;
87 }
88
89 /**
90  * Returns a date as a string yyyy/mm/dd hh:mm:ss UT
91  * @param jd The Julian day
92  * @return The String
93  */
94 public static String getDateAsString(double jd) {
95     if (jd == -1) return "NO RISE/SET/TRANSIT FOR THIS OBSERVER/DATE";
96
97     JulianDay julDay = new JulianDay(jd);
98     double[] hms = Util.getHMS(julDay.getDayFraction() * Constant.TWO_PI);
99     String out = Util.fmt02(julDay.year, "/");
100    out += Util.fmt02(julDay.month, "/");
101    out += Util.fmt02(julDay.day, " ");
102    out += Util.fmt02((int) hms[0], ":");
103    out += Util.fmt02((int) hms[1], ":");
104    out += Util.fmt02((int) (hms[2] + 0.5), " UT");
105    return out;
106 }

```

3. Reducción de coordenadas

Esta es la parte principal del código, que permite la reducción de coordenadas para obtener la posición del objeto desde un observador en la Tierra, a partir de la posición media en coordenadas eclípticas, referida al equinoccio medio de la fecha. El proceso principal se encuentra en el método *doCalc*, en el cual se proporcionan esas coordenadas y se especifica si la reducción debe ser geocéntrica (para un observador hipotético situado en el centro de la Tierra), o referida a un observador real situado en la superficie de la Tierra (topocéntrica). En general, la reducción será siempre topocéntrica, salvo en cálculos específicos como obtener los instantes de los equinoccios y solsticios, o las fases lunares.

El método *doCalc* primero convierte las coordenadas medias en verdaderas sumando la nutación en longitud eclíptica. A continuación, las coordenadas son corregidas por la posición del observador en la Tierra, teniendo en cuenta tanto la forma de geoide de la Tierra como la altura del observador sobre la superficie (variables *geocLat*, *geocR*, y *eradius*). Las coordenadas son después corregidas por aberración diurna, un efecto menor, aunque relevante si queremos obtener posiciones precisas, con errores menores a 0.1". A continuación, se obtienen las coordenadas horizontales de acimut y altura, rotando directamente las coordenadas sin utilizar el código del fichero *CoordinateSystem*, donde la rotación requeriría de dos pasos y no se ahorrarían líneas de código. El código presente después permite corregir la altura por refracción, y obtener los instantes de salida y puesta del objeto, suponiendo que sea estático en el cielo. En estos cálculos se han introducido nuevas constantes en el fichero *Constant.java*: *SPEED_OF_LIGHT* (299792458.0 m/s), y *SIDEREAL_DAY_LENGTH* (1.00273781191135448 días). En el repositorio aparecen comentadas en la clase Java mencionada.

En cuanto a la corrección por refracción, es interesante notar que las fórmulas clásicas como la de Bennet permiten obtener la altura geométrica a partir de la aparente, pues estas fórmulas se derivaron a partir de observaciones en las que lógicamente se medía la altura aparente. Dado que en las efemérides se obtiene la altura geométrica, es necesario invertir numéricamente la fórmula para aplicarla correctamente. Como detalle, en el método *getGeometricElevation* se muestra comentado un código que permitiría corregir por refracción en el caso de que las observaciones se hagan en longitudes de onda radio, en vez de en el óptico. En estos cálculos el efecto de depresión del horizonte no se considera, de manera que la altura proporcionada se refiere al horizonte astronómico.

El método *getBodyPosition* no contiene código en este caso y devuelve un array nulo. La idea es que en otras piezas de código que se deriven de este fichero ese método quede reemplazado por el código necesario para obtener la posición de un objeto particular. Para ello se utiliza la propiedad de herencia, como se describe en la sección siguiente.

El método *getEphemeris* es de tipo estático (no es específico de un objeto de esa clase, sino común a todos ellos), presente con la idea de ser llamado desde otros ficheros para hacer todos los cálculos, incluyendo las iteraciones necesarias para obtener la salida y puesta de objetos que se mueven en el cielo. Estas iteraciones requieren de cambiar la fecha de cálculo para seguir el movimiento del objeto, lo que se hace en el método *setUTDate*.

También es necesario mencionar las construcciones de tipo *enum* presentes en el fichero, que incluyen las distintas opciones para seleccionar el tipo de salida y puesta que se calcula (*TWILIGHT*, que selecciona si los resultados se refieren al orto y ocaso civil, náutico, o astronómico), o para qué fecha deben estar referidos estos valores (*TWILIGHT_MODE*, que selecciona si los resultados deben referirse al momento más cercano al instante de cálculo, o al día actual en hora local o UT). El *enum EVENT* sólo se utiliza internamente y no forma parte de las opciones que deben especificarse para el cálculo de efemérides. En el pasado, lenguajes como Fortran que no tienen este tipo de construcciones requerían de valores enteros o *flags* para especificar este tipo de opciones, de manera que podía introducirse cualquier valor, correcto o no, y el código requería de chequeos para lanzar errores en caso necesario. Además, estos valores numéricos carecían de sentido por sí mismos, y era necesario documentar los valores disponibles y su

significado, y leer el código para saber dar el valor correcto. Los campos de tipo *enum* en Java evitan este problema, pues en ellos sólo hay un número limitado de valores aceptables, y todos ellos vienen expresados por un nombre que identifica de qué estamos hablando. Si se intenta poner un valor distinto, directamente el código no podrá ser compilado y ejecutado, de manera que no hacen falta chequeos. Otros lenguajes incluyen construcciones similares al *enum* de Java, por ejemplo Python a partir de la versión 3.

```

1 package journal;
2
3 /**
4  * A class to reduce the position of a body to obtain the ephemerides as visible from
5  * a given observer, include rise/set times.
6  */
7 public class EphemReduction {
8     /** The set of twilights to calculate (types of rise/set events) */
9     public enum TWILIGHT {
10        /** Identifier to compute rise/set times for astronomical twilight (center of
11         the body at -18 degrees of geometrical elevation) */
12        ASTRONOMICAL(-18),
13        /** Identifier to compute rise/set times for nautical twilight (center of the
14         body at -12 degrees of geometrical elevation) */
15        NAUTICAL(-12),
16        /** Identifier to compute rise/set times for civil twilight (center of the body
17         at -6 degrees of geometrical elevation) */
18        CIVIL(-6),
19        /** The standard value of 34' for the refraction at the local horizon */
20        HORIZON_34arcmin(-34.0 / 60.0);
21
22        /** Target elevation of the center of the body in degrees */
23        private double elevation;
24
25        TWILIGHT(double elev) {
26            elevation = elev;
27        }
28    }
29
30    /** Possible options to return the rise/set/transit times */
31    public enum TWILIGHT_MODE {
32        /** Closest events */
33        CLOSEST,
34        /** Compute events for the current date in UT */
35        TODAY_UT,
36        /** Compute events for the current date in LT */
37        TODAY_LT;
38    }
39
40    /** The set of events to calculate (rise/set/transit events) */
41    public enum EVENT { RISE, SET, TRANSIT }
42
43    protected double jd_UT;
44    private double nutLon;
45    protected double obsLon, obsLat, obsAlt;
46    protected double lst;
47    protected TWILIGHT twilight;
48    protected TWILIGHT_MODE twilightMode;
49    protected int timeZone = 0; /** Time zone for option {@linkplain
50     TWILIGHT_MODE#TODAY_LT}, LT-UT, hours. */
51
52    /**
53     * The constructor with the data for the ephemerides reduction process
54     * @param jd_utc
55     * @param lon
56     * @param lat

```

```

53     * @param alt
54     * @param tw
55     * @param twm
56     * @param tz
57     */
58     public EphemReduction(double jd_utc, double lon, double lat, double alt, TWILIGHT
59         tw, TWILIGHT_MODE twm, int tz) {
60         obsLon = lon * Constant.DEG_TO_RAD;
61         obsLat = lat * Constant.DEG_TO_RAD;
62         obsAlt = alt;
63         twilight = tw;
64         twilightMode = twm;
65         timeZone = tz;
66         setUTDate(jd_utc);
67     }
68     /**
69     * Sets the UT date from the provided Julian day and computes the nutation and
70     * sidereal time
71     * @param jd The new Julian day in UT
72     */
73     public void setUTDate(double jd) {
74         this.jd_UT = jd;
75         double[] nut = EarthAngles.nutation(jd);
76         nutLon = nut[0];
77         lst = EarthAngles.localApparentSiderealTime(jd, obsLon);
78     }
79     /**
80     * Compute the position of the body
81     * @param pos Values for the ecliptic longitude, latitude, distance and so on from
82     * previous methods for the specific body
83     * @param geocentric True to return geocentric position. Set this to false
84     * generally
85     * @return The Ephem object with the output position. The rise/set/transit times
86     * returned here are only valid for non-moving bodies
87     */
88     public EphemData doCalc(double[] pos, boolean geocentric) {
89         // Correct for nutation in longitude
90         pos[0] = pos[0] + nutLon;
91
92         // Ecliptic to equatorial coordinates using true obliquity
93         double[] xyz =
94             CoordinateSystem.eclipticToEquatorial(CoordinateSystem.sphericalToCartesian(pos[0],
95                 pos[1]), jd_UT);
96
97         // Obtain topocentric rectangular coordinates
98         double geocLat = (obsLat - .1925 * Math.sin(2 * obsLat) * Constant.DEG_TO_RAD);
99         double geocR = 1.0 - Math.pow(Math.sin(obsLat), 2) / 298.257;
100        double eradius = (geocR * Constant.EARTH_RADIUS + obsAlt * 0.001);
101        double radiusAU = eradius / (pos[2] * Constant.AU);
102        if (!geocentric) {
103            double cosLat = Math.cos(geocLat);
104            double[] correction = new double[] {
105                radiusAU * cosLat * Math.cos(lst),
106                radiusAU * cosLat * Math.sin(lst),
107                radiusAU * Math.sin(geocLat)};
108            xyz[0] -= correction[0];
109            xyz[1] -= correction[1];
110            xyz[2] -= correction[2];
111        }
112
113        // Obtain spherical topocentric equatorial coordinates
114        double[] sph = CoordinateSystem.cartesianToSpherical(xyz);
115        double ra = sph[0], dec = sph[1], dist = pos[2] * sph[2];

```

```

111 // Correct the equatorial position by diurnal aberration (< 0.3")
112 if (!geocentric) {
113     double rotRate = Constant.SIDEREAL_DAY_LENGTH * Constant.TWO_PI /
114         Constant.SECONDS_PER_DAY; // rad/s
115     double factor = rotRate * (eradius * 1000.0) / Constant.SPEED_OF_LIGHT; //
116         v/c
117     double ddec = factor * Math.cos(geocLat) * Math.sin(dec) * Math.sin(angh);
118     if (Math.cos(dec) != 0.0) ra += factor * Math.cos(angh) * Math.cos(geocLat)
119         / Math.cos(dec);
120     dec += ddec;
121 }
122 // Hour angle
123 double angh = lst - ra;
124 // Obtain azimuth and geometric alt
125 double sinLat = Math.sin(obsLat);
126 double cosLat = Math.cos(obsLat);
127 double sinDec = Math.sin(dec), cosDec = Math.cos(dec);
128 double h = sinLat * sinDec + cosLat * cosDec * Math.cos(angh);
129 double alt = Math.asin(h);
130 double axz = Math.cos(angh) * sinLat - sinDec * cosLat / cosDec;
131 double azi = Math.PI + Math.atan2(Math.sin(angh), axz); // 0 = north
132
133 if (geocentric) return new EphemData(azi, alt, -1, -1, -1, -1,
134     Util.normalizeRadians(ra), dec, dist, pos[0], pos[1], pos[3]);
135 // Get apparent elevation
136 alt = getApparentElevation(alt);
137
138 double tmp = twilight.elevation * Constant.DEG_TO_RAD;
139 // Consider the angular radius (pos[3]) for rise, set, transit times when using
140 // the HORIZON_34arcmin twilight.
141 // Removing angular radius here would do calculations for the center of the
142 // disk instead of the lower/upper limb.
143 if (twilight == TWILIGHT.HORIZON_34arcmin) tmp = tmp - pos[3];
144 // Compute cosine of hour angle
145 tmp = (Math.sin(tmp) - sinLat * sinDec) / (cosLat * cosDec);
146 // Make calculations for the meridian
147 double transit_alt = Math.asin(sinDec * sinLat + cosDec * cosLat);
148 transit_alt = getApparentElevation(transit_alt);
149 // Obtain the current transit event in time
150 double transit = getTwilightEvent(ra, 0);
151 // Make calculations for rise and set
152 double rise = -1, set = -1;
153 if (Math.abs(tmp) <= 1.0) {
154     double ang_hor = Math.abs(Math.acos(tmp));
155     rise = getTwilightEvent(ra, -ang_hor);
156     set = getTwilightEvent(ra, ang_hor);
157 }
158 EphemData out = new EphemData(azi, alt, rise, set, transit, transit_alt,
159     Util.normalizeRadians(ra), dec, dist, pos[0], pos[1], pos[3]);
160 return out;
161 }
162 private double getTwilightEvent(double ra, double angh) {
163     double celestialHoursToEarthTime = 1.0 / (Constant.SIDEREAL_DAY_LENGTH *
164         Constant.TWO_PI);
165     double jdToday_UT = Math.floor(jd_UT - 0.5) + 0.5;

```

```

169 double eventTime = celestialHoursToEarthTime * Util.normalizeRadians(ra + angh
170 - lst);
171 double eventTimePrev = celestialHoursToEarthTime * (Util.normalizeRadians(ra +
172 angh - lst) - Constant.TWO_PI);
173 double eventDatePrev_UT = Math.floor(jd_UT + eventTimePrev - 0.5) + 0.5;
174
175 if (Math.abs(eventTimePrev) < Math.abs(eventTime) && twilightMode ==
176 TWILIGHT_MODE.CLOSEST) eventTime = eventTimePrev;
177 if (twilightMode == TWILIGHT_MODE.TODAY_UT) {
178 double eventDate_UT = Math.floor(jd_UT + eventTime - 0.5) + 0.5;
179 if (jdToday_UT != eventDate_UT) eventTime = -jd_UT - 1;
180 if (jdToday_UT == eventDatePrev_UT) eventTime = eventTimePrev;
181 }
182 if (twilightMode == TWILIGHT_MODE.TODAY_LT) {
183 double tz = timeZone / 24.0, jdToday_LT = Math.floor(jd_UT + tz - 0.5) + 0.5;
184 double eventDate_LT = Math.floor(jd_UT + tz + eventTime - 0.5) + 0.5;
185 if (jdToday_LT != eventDate_LT) eventTime = -jd_UT - 1;
186
187 double eventDatePrev_LT = Math.floor(jd_UT + tz + eventTimePrev - 0.5) + 0.5;
188 if (jdToday_LT == eventDatePrev_LT) eventTime = eventTimePrev;
189
190 double eventTimeNext = celestialHoursToEarthTime * (Util.normalizeRadians(ra
191 + angh - lst) + Constant.TWO_PI);
192 double eventDateNext_LT = Math.floor(jd_UT + tz + eventTimeNext - 0.5) + 0.5;
193 if (jdToday_LT == eventDateNext_LT) eventTime = eventTimeNext;
194 }
195 }
196 return jd_UT + eventTime;
197 }
198
199 /** Corrects input geometric elevation for refraction if it is greater than -3
200 degrees, returning the apparent elevation */
201 private double getApparentElevation(double alt) {
202 if (alt <= -3 * Constant.DEG_TO_RAD) return alt;
203
204 double altIn = alt, prevAlt = alt;
205 int niter = 0;
206 do {
207 double altOut = getGeometricElevation(alt);
208 alt = altIn - (altOut - alt);
209 niter++;
210 if (Math.abs(prevAlt - alt) < 0.001 * Constant.DEG_TO_RAD) break;
211 prevAlt = alt;
212 } while (niter < 8);
213
214 return alt;
215 }
216
217 /** Compute geometric elevation from apparent elevation. Note ephemerides
218 calculates geometric elevation, so an inversion is
219 * required, something achieved in method {@linkplain
220 #getApparentElevation(double)} by iteration */
221 private double getGeometricElevation(double alt) {
222 double ps = 1010; // Pressure in mb
223 double ts = 10 + 273.15; // Temperature in K
224 double altDeg = alt * Constant.RAD_TO_DEG;
225
226 // Bennet 1982 formulae for optical wavelengths, do the job but not accurate
227 close to horizon
228 double r = Constant.DEG_TO_RAD * Math.abs(Math.tan(Constant.PI_OVER_TWO -
229 (altDeg + 7.31 / (altDeg + 4.4)) * Constant.DEG_TO_RAD)) / 60.0;
230 double refr = r * (0.28 * ps / ts);
231 return Math.min(alt - refr, Constant.PI_OVER_TWO);
232 }
233
234

```

```

225     /*
226     // Bennet formulae adapted to radio wavelenths. Use this for position in radio
227     // Reference for some values:
228     // http://icts-yebes.oan.es/reports/doc/IT-OAN-2003-2.pdf (Yebes 40m
229     // radiotelescope)
230     double hs = 20; // Humidity %
231     // Water vapor saturation pressure following Crane (1976), as in the ALMA
232     // memorandum
233     double esat = 6.105 * Math.exp(25.22 * (ts - 273.15) / ts) + Math.pow(ts /
234     273.15, -5.31);
235     double Pw = hs * esat / 100.0;
236
237     double R0 = (16.01 / ts) * (ps - 0.072 * Pw + 4831 * Pw / ts) *
238     Constant.ARCSEC_TO_RAD;
239     double refr2 = R0 * Math.abs(Math.tan(Constant.PI_OVER_TWO - (altDeg + 5.9 /
240     (altDeg + 2.5)) * Constant.DEG_TO_RAD));
241     return Math.min(alt - refr, Constant.PI_OVER_TWO);
242     */
243 }
244
245 /**
246 * Computes an accurate rise/set/transit time for a moving object
247 * @param riseSetJD Start date for the event
248 * @param index Event identifier
249 * @param niter Maximum number of iterations
250 * @return The Julian day in UT for the event, 1s accuracy
251 */
252 protected double obtainAccurateRiseSetTransit(double riseSetJD, EVENT index, int
253 niter) {
254     double step = -1;
255     for (int i = 0; i < niter; i++) {
256         if (riseSetJD == -1) return riseSetJD; // -1 means no rise/set from that
257         // location
258         setUTDate(riseSetJD);
259         EphemData out = doCalc(getBodyPosition(), false);
260
261         double val = out.rise;
262         if (index == EVENT.SET) val = out.set;
263         if (index == EVENT.TRANSIT) val = out.transit;
264         step = Math.abs(riseSetJD - val);
265         riseSetJD = val;
266         if (step <= 1.0 / Constant.SECONDS_PER_DAY) break; // convergency reached
267     }
268     if (step > 1.0 / Constant.SECONDS_PER_DAY) return -1; // did not converge =>
269     // without rise/set/transit in this date
270     return riseSetJD;
271 }
272
273 protected double[] getBodyPosition() {
274     return null;
275 }
276
277 /**
278 * Computes the ephemerides for a body, including accurate rise/set/transit times
279 * @param bodyRed The body reduction data
280 * @return The ephemerides data
281 */
282 public static EphemData getEphemeris(EphemReduction bodyRed) {
283     double jd_UT = bodyRed.jd_UT;
284     EphemData bodyData = bodyRed.doCalc(bodyRed.getBodyPosition(), false);
285     int niter = 15; // Number of iterations to get accurate rise/set/transit times
286     bodyData.rise = bodyRed.obtainAccurateRiseSetTransit(bodyData.rise, EVENT.RISE,
287     niter);

```

```

278     bodyData.set = bodyRed.obtainAccurateRiseSetTransit(bodyData.set, EVENT.SET,
279         niter);
280     bodyData.transit = bodyRed.obtainAccurateRiseSetTransit(bodyData.transit,
281         EVENT.TRANSIT, niter);
282     if (bodyData.transit == -1) {
283         bodyData.transitElevation = 0;
284     } else {
285         // Update Sun's maximum elevation
286         bodyRed.setUTDate(bodyData.transit);
287         bodyData.transitElevation = bodyRed.doCalc(bodyRed.getBodyPosition(),
288             false).transitElevation;
289     }
290     bodyRed.setUTDate(jd_UT);
291     return bodyData;
292 }

```

4. Efemérides aproximadas del Sol

El código presentado en esta sección permite obtener la posición aproximada del Sol para un observador. Para que el lector pueda comprobar los resultados, el método *main* presente al final del listado contiene un ejemplo de cálculo junto con el resultado esperado.

El código utiliza herencia para poder utilizar el código contenido en la clase *EphemReduction*, descrita en la sección anterior, y poder así acceder a los algoritmos de reducción. Esta propiedad también puede utilizarse en otros lenguajes como Python. En el caso de que el lector utilice un lenguaje que no ofrezca esta característica, la opción más sencilla es tener las funciones de reducción como métodos externos que se puedan llamar para calcular las efemérides de cualquier cuerpo, o bien copiar y pegar el código del fichero *EphemReduction.java* dentro de cada uno de los ficheros en los que se calcularán las efemérides de los distintos objetos. La última opción conviene evitarla para no repetir código que debería ser compartido.

La parte esencial es el método *getBodyPosition*, que proporciona la longitud eclíptica, la latitud eclíptica (para el Sol se asume que es cero dado que es siempre muy pequeña), la distancia, y el radio angular del Sol, obtenidos para el equinoccio medio de la fecha. Algunos de los algoritmos utilizados para este cálculo provienen del trabajo de S. L. Moshier [2]. Los resultados están corregidos por aberración o tiempo-luz, de manera aproximada.

El método *main* contiene el ejemplo práctico. El objeto de tipo *EphemSunSimple* contienen las condiciones del cálculo de efemérides, las cuales se utilizan en el método estático *EphemReduction.getEphemeris*. Este método hace internamente todas las operaciones necesarias para iterar a la hora de obtener los instantes precisos de salida y puesta del objeto. El objeto de salida es de tipo *EphemData*, descrito al principio, el cual se utiliza después para expresar los resultados mediante su método *toString*, mencionado anteriormente. De esta manera, el código queda bastante limpio y este fichero permite calcular la posición del Sol sin incluir ningún otro código común.

```

1 package journal;
2
3 /**
4  * A class to compute the ephemerides of the Sun. This class uses the code inside
5  * {@linkplain EphemReduction} by inheritance.
6  */
7 public class EphemSunSimple extends EphemReduction {
8     public EphemSunSimple(double jd_utc, double lon, double lat, double alt, TWILIGHT
9         tw, TWILIGHT_MODE twm, int tz) {
10         super(jd_utc, lon, lat, alt, tw, twm, tz);

```

```

10 }
11
12 @Override
13 public double[] getBodyPosition() {
14     double t = EarthAngles.toCenturiesRespectJ2000(jd_UT, true);
15
16     // Mean longitude of Sun corrected for precession, taken from Moshier.
17     // Final accuracy up to +/- 27" or better over many millenia compared to VSOP87
18     double xe = (129597742.283429 * t + 361679.198) + (-5.23e-6 * t - 2.04411e-2) *
19         t * t;
20     xe += ((((((((-8.66e-20 * t - 4.759e-17) * t + 2.424e-15) * t + 1.3095e-12) *
21         t + 1.7451e-10) * t - 1.8055e-8) * t - 0.0000235316) * t + 0.000076) * t +
22         1.105414) * t + 5028.791959) * t;
23     double lon = Util.normalizeRadians(Math.PI + Constant.ARCSEC_TO_RAD * xe) *
24         Constant.RAD_TO_DEG;
25
26     /* Mean anomaly of sun = l' (J. Laskar) */
27     double x = (1.2959658102304320e+08 * t + 1.2871027407441526e+06);
28     x += (((((((((1.62e-20 * t - 1.0390e-17) * t - 3.83508e-15) * t + 4.237343e-13)
29         * t + 8.8555011e-11) * t - 4.77258489e-8) * t - 1.1297037031e-5) * t +
30         8.7473717367324703e-05) * t - 5.5281306421783094e-01) * t * t;
31     double sanomaly = Util.normalizeRadians(Constant.ARCSEC_TO_RAD * x);
32
33     double c = (1.9146 - .004817 * t - .000014 * t * t) * Math.sin(sanomaly);
34     c = c + (.019993 - .000101 * t) * Math.sin(2 * sanomaly);
35     c = c + .00029 * Math.sin(3.0 * sanomaly); // Correction to the mean ecliptic
36         longitude
37
38     // Now compute approximate aberration
39     double d = -.00569;
40
41     double slongitude = lon + c + d; // apparent longitude (error<0.003 deg)
42     double slatitude = 0; // Sun's ecliptic latitude is always negligible
43     double ecc = .016708617 - 4.2037E-05 * t - 1.236E-07 * t * t; // Eccentricity
44     double v = sanomaly + c * Constant.DEG_TO_RAD; // True anomaly
45     double sdistance = 1.000001018 * (1.0 - ecc * ecc) / (1.0 + ecc * Math.cos(v));
46         // In UA
47
48     return new double[] {slongitude * Constant.DEG_TO_RAD, slatitude *
49         Constant.DEG_TO_RAD, sdistance, Math.atan(696000.0 / (sdistance *
50         Constant.AU))};
51 }
52
53 /**
54  * Test program
55  * @param args Not used
56  */
57 public static void main(String[] args) {
58     // Prepare input data
59     int year = 2020, month = 6, day = 9, h = 18, m = 0, s = 0;
60     JulianDay jday = new JulianDay(year, month, day);
61     jday.setDayFraction((h + m / 60.0 + s / 3600.0) / 24.0);
62
63     double jd_utc = jday.getJulianDay();
64     double lon = -4; // degrees
65     double lat = 40;
66     double alt = 0; // m
67     int tz = 3; // h
68     TWILIGHT tw = TWILIGHT.HORIZON_34arcmin;
69     TWILIGHT_MODE twm = TWILIGHT_MODE.TODAY_UT;
70
71     // Compute the ephemerides data
72     EphemSunSimple sunEph = new EphemSunSimple(jd_utc, lon, lat, alt, tw, twm, tz);
73     EphemData sunData = EphemReduction.getEphemeris(sunEph);
74 }

```

```

65     // Report
66     System.out.println("Sun");
67     System.out.println(sunData.toString());
68
69     /*
70     Sun
71     Az:      285.78873°
72     El:      17.424834°
73     Dist:    1.015206 au
74     RA:      78.41086°
75     DEC:     23.007696°
76     Ill:     100.0%
77     ang.R:   0.26256925°
78     Rise:    2020/06/09 04:46:57 UT
79     Set:     2020/06/09 19:43:59 UT
80     Transit: 2020/06/09 12:15:22 UT (elev. 72.99493°)
81     */
82     }
83 }

```

5. Posición aparente de estrellas y otros objetos de catálogos

Las coordenadas de estrellas y objetos de cielo profundo presentes en los catálogos son fijas, al margen de posibles movimientos propios que puedan tener esos objetos a lo largo de los siglos. Estas coordenadas son también medias, como las calculadas en los métodos anteriores, pero están referidas a un equinoccio particular, habitualmente el equinoccio 2000. Para obtener las efemérides de una estrella primero debemos corregir la posición del catálogo por precesión, y después aplicar el método de reducción.

Para ello, una vez instanciado el objeto *EphemStar* en el ejemplo, equivalente al *EphemSunSimple* de la sección anterior, se utiliza el método *setJ2000Position* para establecer las coordenadas de catálogo del objeto, las cuales están incluidas como variables privadas en la clase Java. Por consistencia y claridad, el constructor no ha sido modificado respecto a códigos anteriores, aunque se podría incluir la introducción de esas coordenadas en el constructor y evitar la necesidad del método *setJ2000Position*. El método *getBodyPosition* devuelve las coordenadas de la estrella, corrigiendo primero por precesión mediante el método *equatorialJ2000ToMeanEquatorialOfDate*, que contiene un código idéntico al mostrado en una entrega anterior de esta sección [3] para la clase *CoordinateSystem*, pero adaptado para la fecha introducida en el constructor. Este código debe estar lógicamente implementado previamente, y disponible con ese nombre de fichero en el mismo directorio o paquete, junto con el resto del código presentado en entregas anteriores.

El ejemplo implementa el cálculo de la posición aparente de la estrella Vega, a partir de su posición J2000 18h 36m 56s, 38° 47' 00". La posición aparente se muestra al ejecutar el programa, y puede compararse con la proporcionada en el portal *theskylive.com*. No es posible cuantificar con claridad la diferencia que se obtiene en este programa con respecto al valor proporcionado en esa página web [4], sólo puede decirse que debe estar claramente por debajo del segundo de arco.

```

1 package journal;
2
3 /**
4  * A class to compute the ephemerides of a star, or any other catalog body.
5  * This class uses the code inside {@linkplain EphemReduction} by inheritance.
6  */
7 public class EphemStar extends EphemReduction {
8
9     private double ra;
10    private double dec;
11

```

```

12 public EphemStar(double jd_utc, double lon, double lat, double alt, TWILIGHT tw,
13     TWILIGHT_MODE twm, int tz) {
14     super(jd_utc, lon, lat, alt, tw, twm, tz);
15 }
16
17 /**
18  * Sets the J2000 position of the body from catalog coordinates
19  * @param ra Right ascension, hours
20  * @param dec Declination, degrees
21  */
22 public void setJ2000Position(double ra, double dec) {
23     this.ra = ra * 15.0 * Constant.DEG_TO_RAD;
24     this.dec = dec * Constant.DEG_TO_RAD;
25 }
26
27 /**
28  * Transforms coordinates from J2000 equatorial to mean equinox of date
29  * @param p Rectangular coordinates
30  * @return Mean equatorial rectangular coordinates
31  */
32 public double[] equatorialJ2000ToMeanEquatorialOfDate(double[] p) {
33     double t = EarthAngles.toCenturiesRespectJ2000(jd_UT, true);
34     double[] pa = EarthAngles.precessionAnglesFromJ2000(Constant.J2000 + t *
35         Constant.JULIAN_DAYS_PER_CENTURY);
36     return
37         CoordinateSystem.rotate(CoordinateSystem.rotate(CoordinateSystem.rotate(p,
38             CoordinateSystem.getRotZ(-pa[1])), CoordinateSystem.getRotY(pa[2])),
39             CoordinateSystem.getRotZ(-pa[0]));
40 }
41
42 @Override
43 public double[] getBodyPosition() {
44     double[] cartesianJ2000 = CoordinateSystem.sphericalToCartesian(ra, dec);
45     double[] meanEq = equatorialJ2000ToMeanEquatorialOfDate(cartesianJ2000);
46     double[] meanEcl = CoordinateSystem.equatorialToEcliptic(meanEq, jd_UT);
47     double[] meanEclSph = CoordinateSystem.cartesianToSpherical(meanEcl);
48     return new double[] {meanEclSph[0], meanEclSph[1], 1E100, 0}; // Assume
49         infinite distance and angular size 0
50 }
51
52 /**
53  * Test program
54  * @param args Not used
55  */
56 public static void main(String[] args) {
57     // Julian day for current instant
58     JulianDay jday = new JulianDay(System.currentTimeMillis());
59
60     // Observer position (Madrid, as in theskylive.com)
61     double lon = -(3 + 42 / 60.0 + 9.2 / 3600.0); // degrees
62     double lat = 40 + 24 / 60.0 + 59.4 / 3600.0;
63     double alt = 0; // m
64     int tz = 2; // h
65
66     // Catalog body, J2000 coordinates. Check Vega with
67     // https://theskylive.com/sky/stars/vega-alpha-lyrae-star
68     String name = "Vega";
69     double ra = 18.0 + 36 / 60.0 + 56 / 3600.0;
70     double dec = 38 + 47 / 60.0 + 0 / 3600.0;
71
72     // Compute the ephemerides data
73     EphemStar bodyEph = new EphemStar(jday.getJulianDay(), lon, lat, alt,
74         TWILIGHT.HORIZON_34arcmin, TWILIGHT_MODE.TODAY_UT, tz);
75     bodyEph.setJ2000Position(ra, dec);
76     EphemData bodyData = EphemReduction.getEphemeris(bodyEph);

```

```
69
70 // Report
71 System.out.println(name);
72 System.out.println(bodyData.toString());
73
74 System.out.println(" RA: " + Util.formatRA(bodyData.rightAscension, 1));
75 System.out.println(" DEC: " + Util.formatDEC(bodyData.declination, 0));
76 }
77 }
```

References

- [1] Repositorio de código en GitHub, <https://github.com/JCAAC-FAAE>
- [2] Algoritmos de S. L. Moshier: <https://www.moshier.net/>
- [3] *Astronomical computing: Cálculo de ángulos de orientación de la Tierra y transformaciones de coordenadas*, T. Alonso Albi, JCAAC **2**, 87 (2025).
- [4] Posición aparente de la estrella Vega: <https://theskylive.com/sky/stars/vega-alpha-lyrae-star>. Para comparar con el programa es necesario introducir antes la posición del observador (Madrid en el ejemplo del programa)

SECTION: SOFTWARE FOR PHOTOMETRY & ASTROMETRY

Fotometría con Tycho: fotometría de cometas multiapertura y AfRho a partir de un *Track Stack*

Ramón Naves¹ and Montse Campàs²

¹Obs. Montcabrer - MPC 213, Cabriils–Barcelona, Spain. E-mail: ramonnavesnogues@gmail.com.

²Obs. Montcabrer - MPC 213, Cabriils–Barcelona, Spain. E-mail: mcampast@gmail.com.

Keywords: astrometría, fotometría, Tycho

© Este artículo está protegido bajo una licencia [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/)

Abstract

Este artículo pretende ofrecer un tutorial cuya intención es explicar cómo usar Tycho para obtener resultados fotométricos para cometas compatibles con los que se obtienen con los programas *Astrometrica* y *Focas*.

Abstract

This article aims to provide a tutorial intended to explain how to use Tycho to obtain photometric results for comets compatible with those obtained using the software *Astrometrica* together with *Focas*.

1. Introducción

Antes de adentrarnos en este campo sería útil conocer los antecedentes de la Fotometría de cometas y que tiene de especial, frente a la fotometría estelar. La fotometría de cometas tiene algunas peculiaridades. En fotometría estelar es muy frecuente que existan cartas estelares para las variables más estudiadas, con las estrellas de calibrado y control ya elegidas previamente, para que los diversos observadores usen, esas mismas estrellas, en la medida de lo posible. Pero claro, los cometas se mueven por el cielo, por lo que no existen unas cartas fotométricas asociadas a un cometa en particular. Otro problema es que los cometas son difusos, tienen cierta extensión, no son objetos puntuales como las estrellas, así que la elección de la caja o box fotométrico adecuado también supone un problema añadido. Asimismo, las estrellas tienen su propio brillo, y sobre todo un amplio abanico de colores. En cambio podemos considerar que los cometas, como solo reflejan la luz solar y no tienen luz propia, tienen todos el mismo color, prácticamente el color del Sol. Aunque es bien conocido que los grandes cometas emiten su propia luz verdoso azulada por fluorescencia al ser iluminados sus compuestos gaseosos por los rayos ultravioletas del Sol. Aunque esta luz verdosa solo supone una pequeña contribución al brillo total de cometa.

Por tanto, aunque es una simplificación, consideraremos que en general los cometas tienen todos el mismo color que el Sol, ello nos permite hacer fotometría sin usar filtros fotométricos, sin que ello suponga un gran disparate, como si lo sería para hacer fotometría sin filtros de estrellas variables, donde cada estrella tiene su propio color. Con estas tres premisas en mente, hace unos 25 años, el Grupo de Observadores de cometas propuso ideas para intentar solventar el problema de la dispersión en los datos fotométricos que se obtenían hasta entonces. Primero se decidió fijar un box fotométrico que fue de 10x10 segundos de arco. Se eligió un box cuadrado porque el software que conocíamos entonces CCDOPS, es lo que permitía. Los primeros resultados mostraron que la dispersión de los datos obtenidos

había bajado bastante, pero seguía siendo insuficiente. Así que decidimos que no nos podíamos fiar de elegir una o dos estrellas de calibrado, porque los resultados cambiaban enormemente dependiendo de las estrellas elegidas.

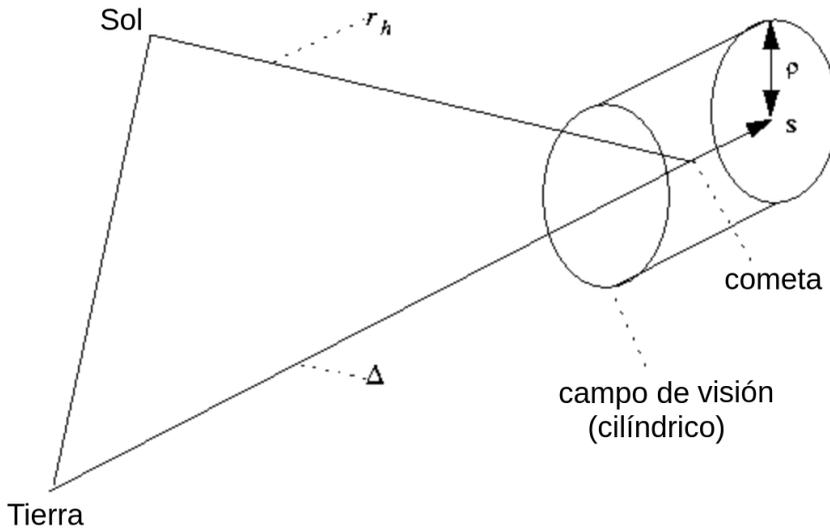


Figura 1. Geometría para la definición del parámetro $Af(\rho)$. $\rho = \rho$ es el radio del cilindro.

En ese momento empezamos a probar varios programas que tuvieran la posibilidad de contar con una buena base de datos fotométrica y que además el cálculo de los centroides en las imágenes fuera de lo más exacto posible. Probamos varios programas distintos y después de muchas pruebas, el programa ganador resultó ser “ASTROART”. La idea era elegir todas las estrellas posibles dentro del catálogo USNO 2.0 en la banda R (banda R por aquello de considerar que era el color más apropiado para los cometas por ser más parecido al Sol, junto con la mayor sensibilidad de las cámaras CCD de la época en esa banda.). Elegir las estrellas de comparación con Astroart una a una, era una tarea tediosa y propensa al error.

Por fortuna contábamos con la ayuda de Julio Castellano, quien desarrolló un software llamado Fase 3, que ayudaba en esta cuestión. Fase 3 ayudaba a correlacionar con un par de clics, las estrellas de tu imagen, con las estrellas correspondientes del catálogo USNO. Aunque su trabajo no acababa aquí, filtraba dicho catálogo y siguiendo criterios fotométricos eliminaba las estrellas de calibrado que tenían mayor discrepancia hasta un valor sigma determinado. Cabe recordar que el catálogo USNO 2.0 partía de imágenes de película química escaneada en la banda R, habían estrellas con un error de catálogo que podía superar las dos magnitudes. Hoy en día es un catálogo obsoleto, pero hace 25 años, para nosotros, supuso algo donde agarrarnos. Fase 3 con la ayuda de los centroides de Astroart realizaba la fotometría por su cuenta con la novedad de añadir nuevos boxes fotométricos cuadrados de 10 x10, 20x20, 30x30, 40x40, 50x50 y 60x60 segundos x lado, a esta fotometría la llamamos fotometría Multibox. Fase 3 supuso toda una revolución pues por primera vez obteníamos una fotometría consistente fuera cual fuera el observador.

Pasaron los años y apareció en escena un programa astrométrico maravilloso llamado Astrometrica. El programa realizaba una astrometría muy buena pero la fotometría que realizaba de los cometas no tenía mucha relación con la que obteníamos con Fase 3. De nuevo propusimos a Julio castellano si podía hacer algo parecido a lo que hacía Fase 3 con Astroart, pero esta vez ayudándose de

Astrometrica. Resultó que Astrometrica generaba unos Logs internos que permitieron al nuevo programa (esta vez lo llamamos Focas) calcular la fotometría a partir de los centroides calculados por Astrometrica. Resultó mucho más fácil usar Astrometrica y Focas que usar la versión anterior de Fase 3 con Astroart. En ese momento Julio Castellano cambió la fotometría a unos box circulares, pero que tuvieran el mismo tamaño que los antiguos box cuadrados, para que de esta manera la nueva fotometría fuera compatible con la antigua.

Con la aparición del catálogo CMC 14-15 cambiamos a este último por ser un catálogo realizado con una cámara CCD y filtro fotométrico Sloan r, se añadió un pequeño offset para la compatibilidad con las medidas antiguas. Por último nos pasamos al catálogo Gaia Banda G por ser esta muy similar a la respuesta de un chip CCD sin filtro fotométrico, y por dar unos resultados muy parecidos a los obtenidos anteriormente. Focas también incorporo el cálculo de A_{frho} .

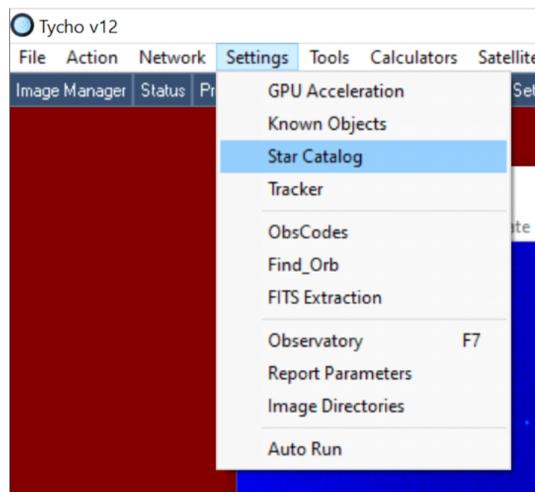


Figura 2. Fotometría con Tycho. Elegir configuración del catálogo.

2. $A_f(\text{Rho})$ - Breve introducción (Julio Castellano Roig - MPC 939)

El parámetro $A_f(\text{Rho})$ [1, 2, 3], ampliamente utilizado en fotometría cometaria, fue introducido en 1984 por Michael A'Hearn con el objetivo de describir el brillo de la coma de los cometas. Se define como el producto del albedo de las partículas, por un "factor de llenado" y por la superficie de la apertura fotométrica empleada proyectada a la distancia del cometa. El factor de llenado sería el cociente entre la sección de las partículas de polvo y la superficie proyectada de la apertura. La cantidad $A_f(\text{rho})$ define la altura de un cilindro de base equivalente a la proyección de la apertura fotométrica, completamente lleno con las partículas de polvo. Usualmente se mide en cm.

2.1. Independencia de la apertura

Los autores del trabajo original y otros, sostienen que el parámetro $A_f(\text{Rho})$ es independiente de la apertura fotométrica empleada. Basta una pequeña reflexión para comprobar que esta afirmación es errónea: implica que el comportamiento de los cometas es uniforme, que la tasa de producción de polvo cambia suavemente y que la velocidad de expansión de este es constante. El cometa 29P es un buen ejemplo de lo contrario. Este cometa se caracteriza por episodios más o menos espaciados de un

súbito aumento de brillo de varias magnitudes. Durante los primeros instantes, una apertura fotométrica pequeña puede abarcar la totalidad del polvo producido, mientras que una más grande no se vería afectada (salvo la parte que comprende a la apertura menor). Conforme el polvo se va expandiendo, llena las aperturas más grandes, que registran un aumento de brillo, mientras que en la más pequeña, la pérdida de densidad induce el descenso del nivel de brillo inicial.

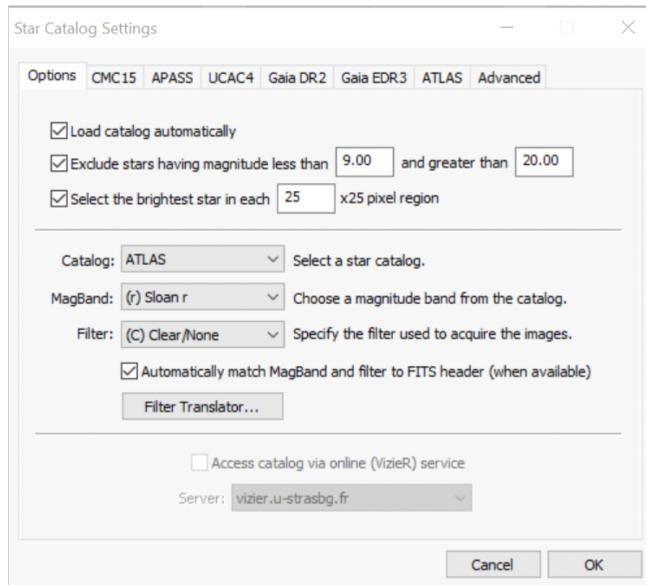


Figura 3. Setting Star Catalog: opciones a rellenar.

2.2. Producción de polvo

Es evidente que el parámetro Af(Rho) está directamente relacionado con la producción instantánea de polvo, sobre todo en aperturas pequeñas, según lo visto en el párrafo anterior. Sin embargo, hay discrepancias importantes entre los distintos autores y resultados consultados, que llegan a superar un factor de 10. En estas condiciones, sería poco prudente establecer una relación determinada entre ambos valores y parece preferible no concretar numéricamente esa dependencia.

3. Fotometría con TYCHO

Para fotometría de cometas, con o sin filtro fotométrico, verificaremos que estamos usando el catálogo Atlas. Usaremos por defecto la banda Sloan r si no estamos usando ningún filtro. Para ello clicamos en Settings Star catalog (Fig. 2).

Lo configuraremos de la siguiente manera (Fig. 3): para fotometría de cometas, con o sin filtros, verificaremos que estamos usando el catálogo Atlas. Configurándolo según el filtro empleado (en este ejemplo elijo la banda Sloan r por ser la que da resultados más parecidos a Gaia). Aunque en este ejemplo concreto elegimos Clear, porque en realidad no se ha usado ningún filtro fotométrico. En caso de usarlos, elegiríamos la banda y el filtro que corresponda con el filtro empleado. Procederemos como si fuéramos a realizar la astrometría del cometa, y para ello cargamos todas las imágenes, Fig. 4. Previamente las habremos astrometrizado, calibrado y alineado con la opción Express mode (ver

artículo anterior en la serie, [4]). Ahora hacemos click en Load Known Objects para que nos aparezca la lista de objetos conocidos.

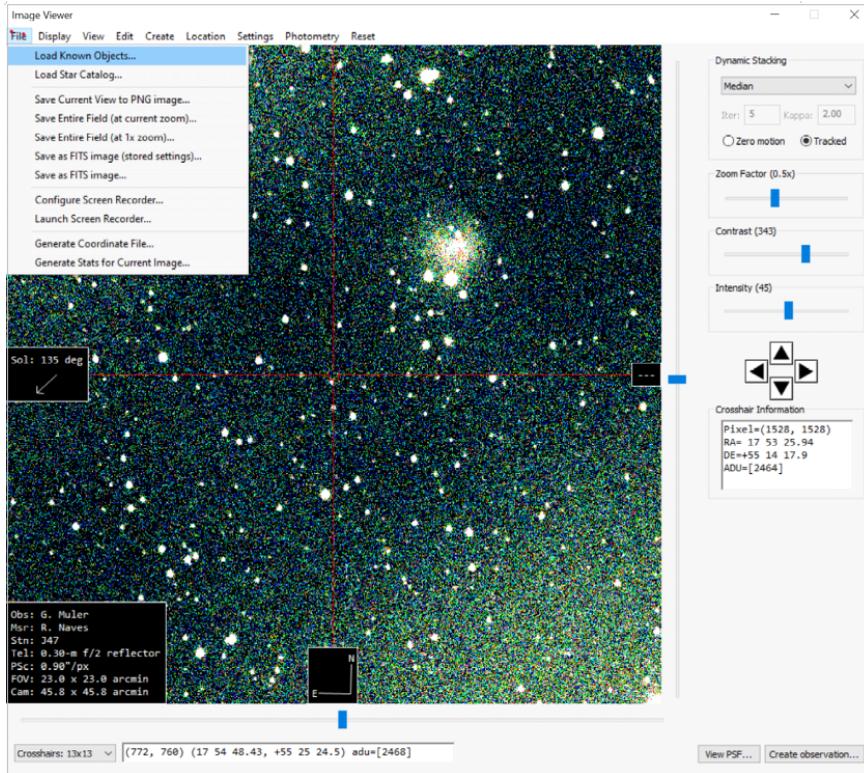


Figura 4. Carga de las imágenes.

A continuación seleccionamos el cometa a medir, en el caso del ejemplo, 12P (Fig. 5), y clicamos en Add To Track Navigator, Fig. 6, y posteriormente en Verify Track, Fig. 7.

Item	Speed	PA	Mag	Dist(AU)	Number	Name	X	Y	RA	DE
1	0.551089	251.04	11.1	3.443	12P	Pons-Brooks	1847	1261	17 52 53.22	+55

Figura 5. Lista de objetos a cargar.

A partir de aquí nos aparece la ventana mostrada en la Fig. 8, donde solo marcaremos un único stack.

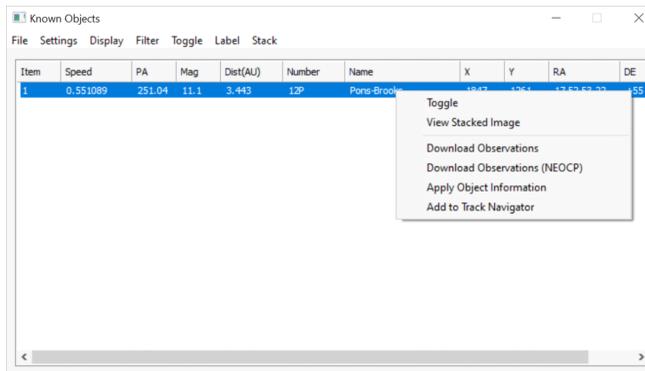


Figura 6. Selección del objeto.

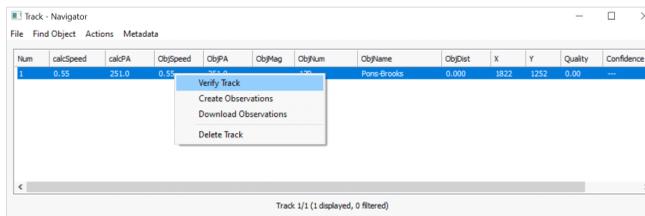


Figura 7. Selección de Verify Track.

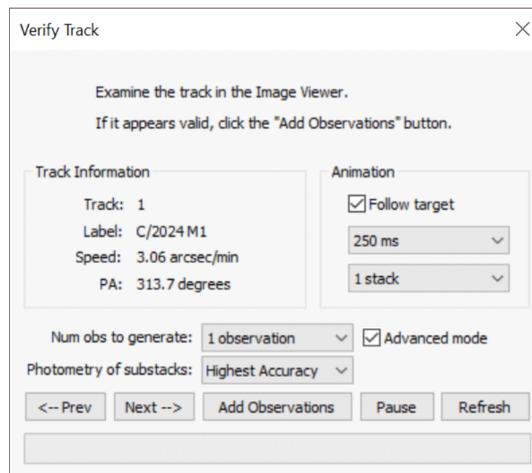


Figura 8. Ventana asociada al menú Verify Track.

Nos aparecerá entonces la imagen sumada sobre el cometa, Fig. 9. Una cuestión importante: en Dynamic Stacking debemos tener puesta la opción Median.

A continuación clicaremos en Settings Overlay, y después en Apply Object. Esta acción nos abrirá un nuevo menú (Fig. 11), donde introduciremos el nombre del cometa, y clicaremos OK. El campo Object Number lo reservaremos solo para los cometas periódicos numerados, y Object Name para los cometas con designación provisional. Clicamos en 12P, y posteriormente en OK para que aparezcan los parámetros del cometa en la imagen. En la ventana Image Viewer nos aparecerá un recuadro con

los datos correspondientes al cometa.

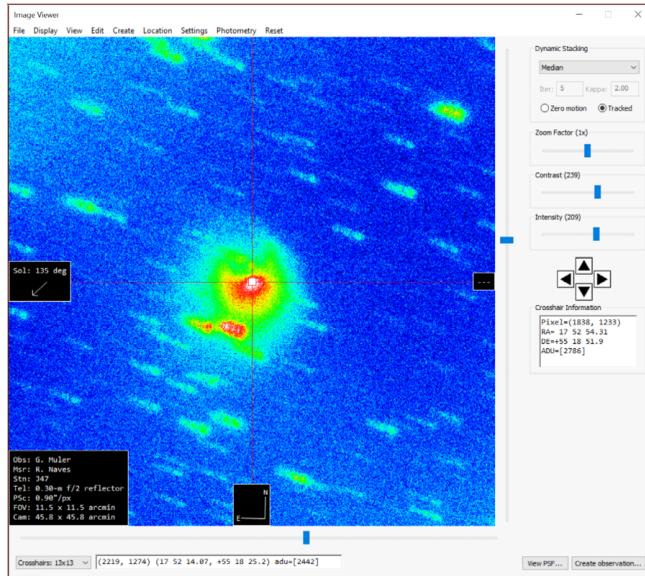


Figura 9. Visor de imágenes con la imagen del cometa.

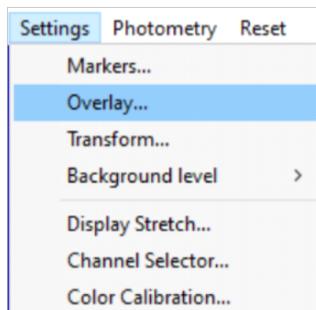


Figura 10. Selección del menú Overlay.

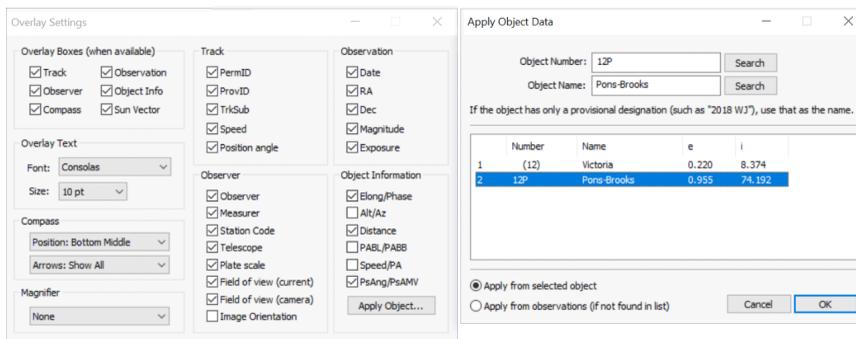


Figura 11. Ventana del menú *Overlay Settings*.

Para realizar la fotometría del cometa, clicamos en *Photometry* y luego en *Comet Photometry* para abrir el correspondiente menú, Fig. 12.

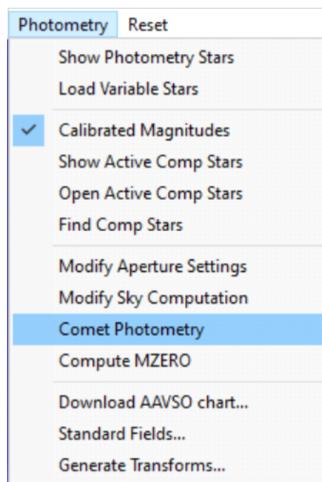


Figura 12. Selección del menú *Comet Photometry*.

Nos aparecerá entonces una ventana, Fig. 13, donde seleccionaremos qué imagen corresponde a cada uno de los menús *Star Image*, *Comet Image*, y *Origin*. En *Star Image* la imagen aparecerá con las estrellas puntuales y el cometa movido. En este paso clicamos *Apply from Image Viewer*, pero antes debemos haber marcado *Zero Motion* en la imagen inferior del campo estelar. *Star Image* se debe corresponder con una imagen suma con las estrellas puntuales, por lo que es imprescindible tener marcado *Zero Motion* en *Image Viewer*. En *Dynamic Stacking* debemos tener elegida la opción *Median*. Si elegimos *Comet Image*, Fig. 14, la imagen aparecerá con las estrellas movidas y el cometa puntual. En este paso clicaremos *Apply from Image Viewer*, pero antes debemos haber marcado *Tracked* en la imagen inferior del campo estelar; *Comet Image* se debe corresponder con una imagen suma con el cometa puntual, por lo que es imprescindible tener marcado previamente esta opción *Image Viewer*.

En el caso de *Origin*, Fig. 15, la imagen aparece con las estrellas movidas y el cometa puntual. En este paso clicamos *Apply from Image Viewer*, pero antes debemos haber marcado *Tracked* en la imagen inferior del campo estelar. *Origin* se debe corresponder con una imagen suma con el cometa

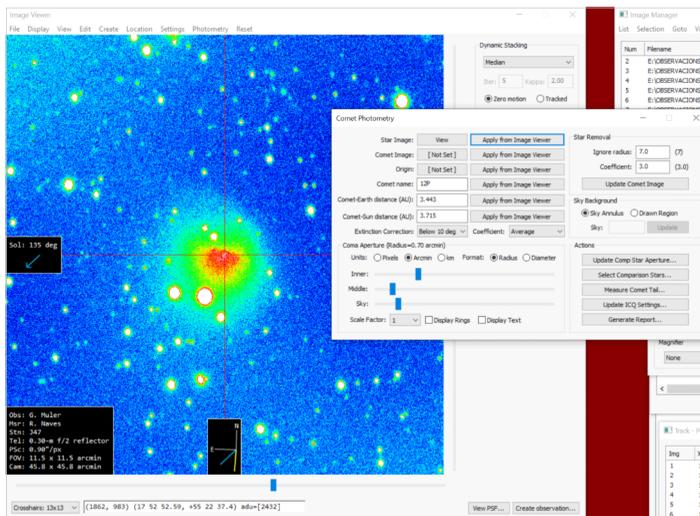


Figura 13. Ventana del menú Comet Photometry sobre la del Image Viewer, con la opción Star Image.

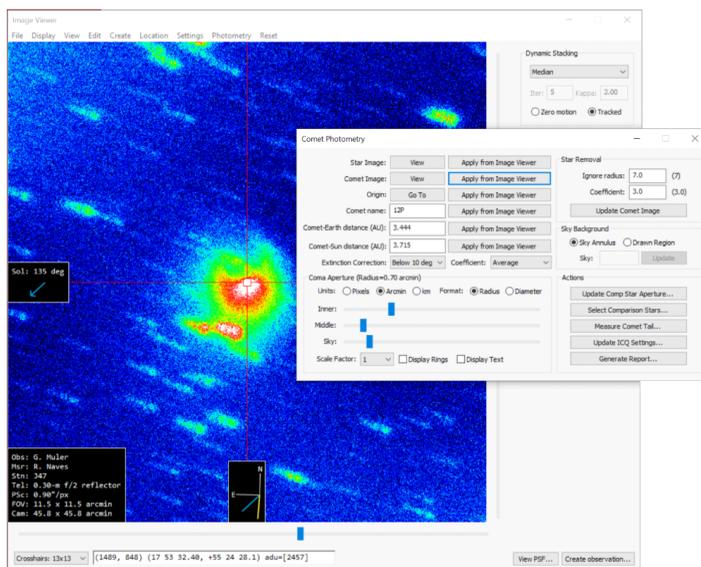


Figura 14. Ventana del menú Comet Photometry sobre la del Image Viewer, con la opción Comet Image.

puntual, que debe salir centrado en el recuadro, por lo que es imprescindible tener marcado previamente Tracked en Imagen View. Clicando Go To el cometa debe aparecer recuadrado.

El programa se encarga de borrar las estrellas para poder obtener una fotometría correcta. En el submenú Sky Background (Fig. 16) marcaremos, con la tecla *shift* apretada, un cuadrado que nos cubra sobradamente la coma del cometa (no es algo crítico pues el resultado no cambia aunque seleccionemos más o menos coma). También podremos activar, marcando dentro del cuadradito Display Rings, los

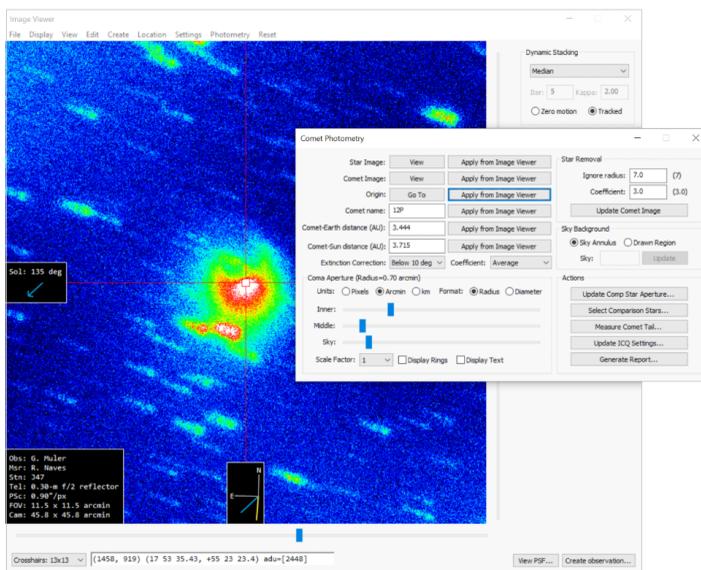


Figura 15. Ventana del menú Comet Photometry sobre la del Image Viewer, con la opción Origin.

anillos concernientes a la coma de cometa. Estos ajustes tampoco afectan a las medidas de Afrho.

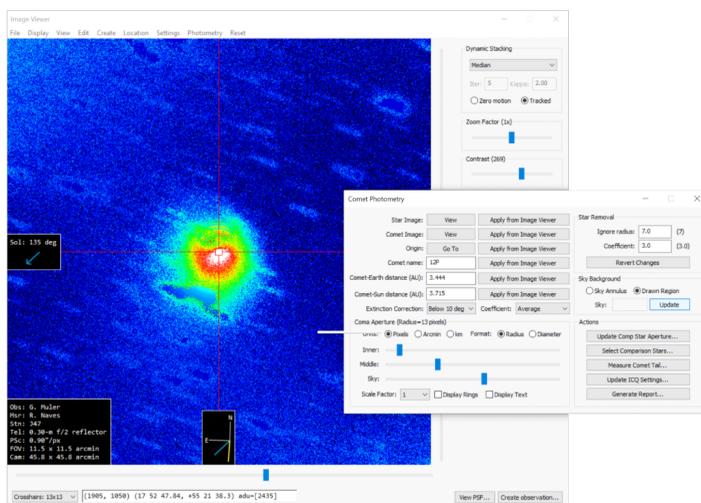


Figura 16. Submenú Sky Background.

En Modify Aperture Settings podemos elegir el tamaño del box fotométrico adecuado, para ser compatibles con los métodos que hemos usado anteriormente hasta ahora (Astrometrica+Focas). Inicialmente (Astroart + Fase 3) usábamos un box fotométrico cuadrado de 10 x 10 segundos, es decir, de 100 segundos de arco al cuadrado. Posteriormente, aunque le seguimos llamando 10 x 10, en realidad en Focas ya usábamos un box circular de 100 segundos cuadrados de superficie. Aquí haremos lo mismo y para ello usaremos box circulares de 100 segundos de arco cuadrados.

100 segundos cuadrados es equivalente a un box circular de 5.64'' de radio, ya que el radio es 88

$(100/\pi)^{1/2} = 5.64$, siendo π el número pi. Ahora bien, Tycho trabaja en pixels, y por tanto habrá que calcular el radio en pixels. Para ello, tendremos que dividir 5.64 por la resolución por pixel a la que trabajamos con nuestra cámara digital. En este ejemplo trabajamos a $0.9''/\text{px}$, y el cálculo a realizar es muy simple:

$$\text{radio en pixels para Tycho} = \frac{5.64}{\text{resolución en ''/px}} = \frac{5.64}{0.9} = 6.26 \text{ px.}$$

Este es el valor que elegiremos para todos los box, Fig. 17. En caso de no conocer la resolución a la que trabajamos, el propio Tycho nos la pone abajo en la imagen, $\text{PSc} = 0.9''/\text{px}$.

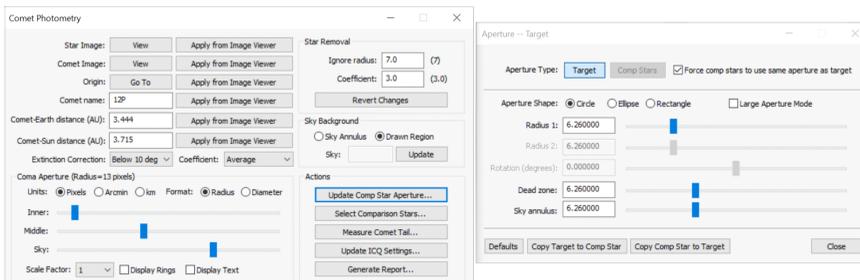


Figura 17. Elección de las aperturas.

El submenú **Select Comparison Stars**, Fig. 17 sirve para elegir las estrellas de comparación siguiendo unos criterios óptimos. Con el catálogo **AtLas** es posible hacerlo como en este ejemplo. Debemos elegir las estrellas que cumplan el criterio elegido marcándolas a partir de la lista que nos aparece, Fig. 18. Es bueno filtrar por SNR; en este ejemplo está solo en el valor 70, pero es preferible siempre que se pueda subirlo a valores más altos como 100 o 200, siempre que aún tengamos suficientes estrellas. Yo recomendaría como mínimo diez estrellas. Podemos filtrar también por magnitud. Se trata de obtener un valor **Std lo** más bajo posible, en este caso 0.092, pero manteniendo un buen número de estrellas. Las estrellas más alejadas de la pendiente deben excluirse. Las estrellas de calibrado se pueden añadir, o bien clicándolas directamente de los puntos rojos, o seleccionándolas de la lista de la izquierda, Fig. 19. Nos aparece una lista con las estrellas de comparación seleccionadas.

Finalmente, clicando en el submenú **Generate Measurements**, nos aparecerán los resultados, como los mostrados en la Fig. 20.

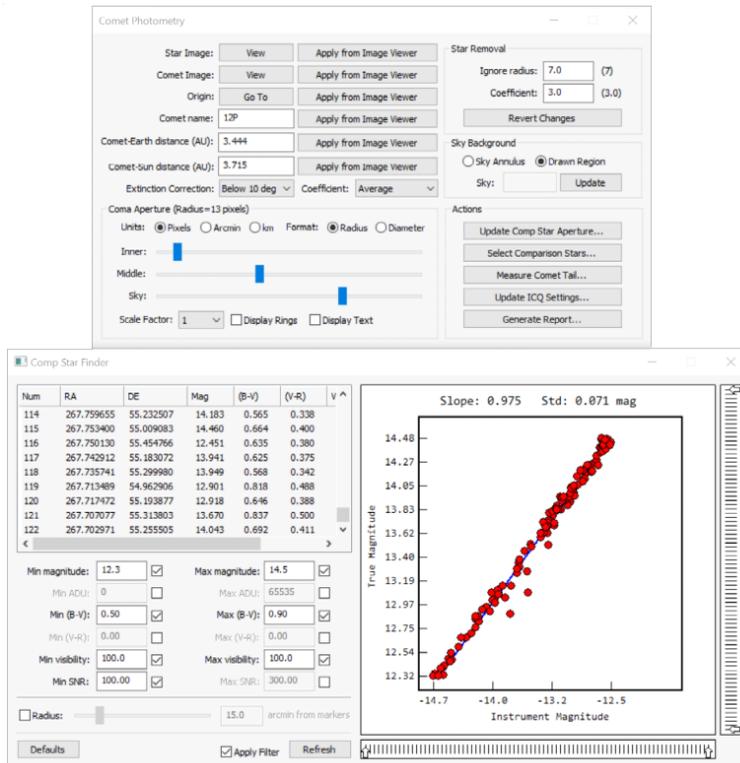


Figura 18. Submenú Select Comparison Stars.

Referencias

- [1] Comet Bowell 1980b, A'Hearn et al. (1984),
http://adsabs.harvard.edu/cgi-bin/nph-data_query?bibcode=1984AJ.....89..579A&db_key=AST&link_type=GIF
- [2] *An Engineering Model of the Dust and Gas Environment of the Inner Coma of Comet P/Wirtane*, M. Müller y E. Grün (1997). ESA, Directorate of Technical and Operational Support, RO-ESC-TA-5501.
- [3] *Le parametre $A_f(\text{Rho})$ - traceur de la activité cometaire*,
<http://perso.wanadoo.fr/fkometes/documents/pdf/ohp111204.pdf>, L. Jorda.
- [4] R. Naves y M. Campàs, JCAAC 2, 105 (2025).

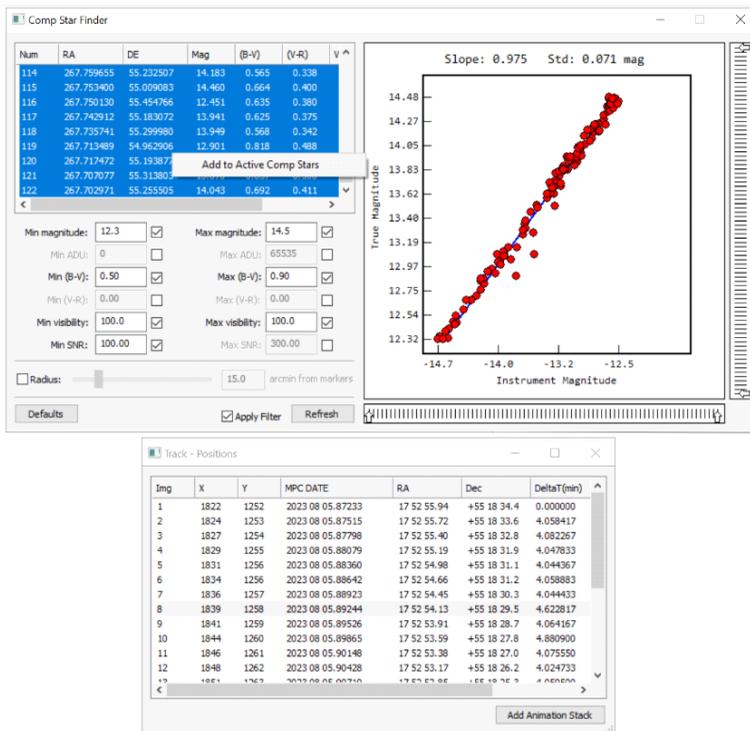


Figura 19. Elección de estrellas de comparación en el submenú Select Comparison Stars.

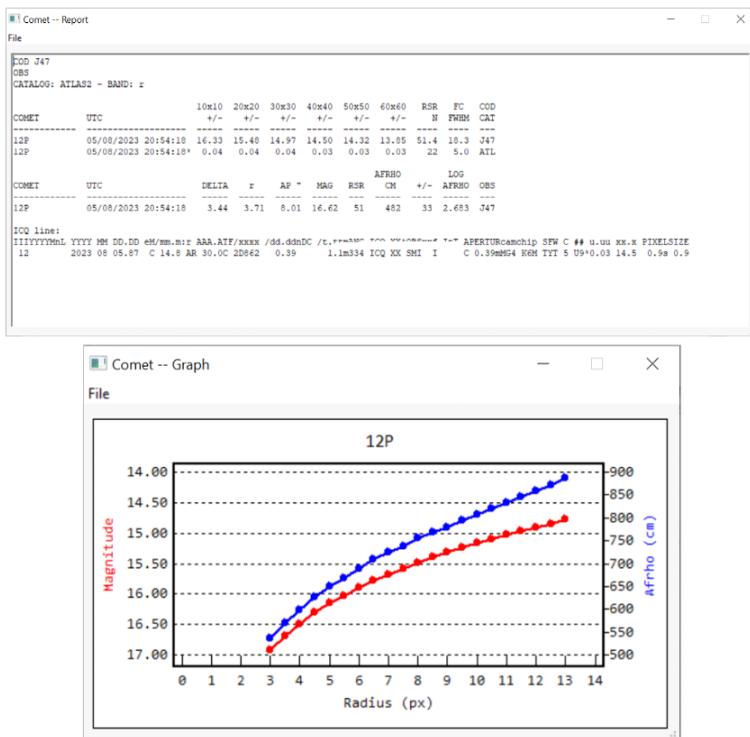


Figura 20. Resultados finales de la fotometría del cometa.

1001011101001100
1111001010011110
1001011101001100
1111001010011110
00e9 4819
fdf0 8888 0098 7667
778a 8828 8888 003c
3b82 0010 0000
0004 0218 7b7a
c7c8 089a ffff

```
import math  
def kepler_solver(eccentricity, mean_anomaly, tolerance=1e-10, max_iterations=1000):  
    eccentric_anomaly = mean_anomaly # Initial guess  
    for _ in range(max_iterations):  
        f = eccentric_anomaly - eccentricity * math.sin(eccentric_anomaly) - mean_anomaly  
        f_prime = 1.0 - eccentricity * math.cos(eccentric_anomaly)  
        eccentric_anomaly -= f / f_prime  
    if abs(f) < tolerance:  
        return eccentric_anomaly
```

